轻松学习最新流行软件系列丛书

轻松学习

Visual Basic 4.0

吴东辉　张　波　编著

清华大学出版社
TSINGHUA UNIVERSITY PRESS

Digitized by the Internet Archive
in 2019 with funding from
Kahle/Austin Foundation

https://archive.org/details/qingsongxuexivis0004unse

轻松学习最新流行软件系列丛书

轻松学习
Visual Basic 4.0

吴东辉　张　波　编著

清华大学出版社

（京）新登字 158 号

内容简介

本书以循序渐进的内容、丰富的图例介绍 Visual Basic 4.0，使读者可以轻松地学习和掌握这一最新流行软件。本书首先介绍了 Windows 95 的特点和基本的操作方法，并介绍了 Visual Basic 的历史和 4.0 版的安装；接着介绍了 Visual Basic 4.0 的编程环境以及如何建立一个简单的应用程序；然后详细说明了窗体和控件的概念和设计方法；解说了 Visual Basic 的基本语言；并讨论了菜单和对话框的设计、图形和绘图操作、文件操作以及错误处理与程序调试技术；最后介绍了 Visual Basic 4.0 的高级程序设计技术，包括 OLE、定制控件和数据库管理等。在附录中，还给出了 Visual Basic 4.0 的术语。

版权所有，翻印必究。
本书封面贴有清华大学出版社激光防伪标签，无标签者不得销售。

图书在版编目（CIP）数据

轻松学习 Visual Basic 4.0/吴东辉，张波编著．－北京：清华大学出版社，1996.2
（轻松学习最新流行软件系列丛书）
ISBN 7-302-02114-7

Ⅰ．轻…　Ⅱ．①吴…　②张…　Ⅲ．Basic 语言-程序设计　Ⅳ．TP312Ba

中国版本图书馆 CIP 数据核字（96）第 03275 号

出版者：清华大学出版社（北京清华大学校内，邮编 100084）
印刷者：国防工业出版社印刷厂
发行者：新华书店总店北京科技发行所
开　本：787×1092　1/16　印张：23.5　字数：582 千字
版　次：1996 年 4 月第 1 版　1997 年 8 月第 4 次印刷
书　号：ISBN 7-302-02114-7/TP・993
印　数：22001～28000
定　价：25.00 元

目　　录

前言 …………………………………………………………………… Ⅰ

第一章　Windows 95 入门 …………………………………… 1

1.1　Microsoft Windows 95 的概述及安装 ……………………… 1

1.1.1　Windows 95 概述 ……………………………………… 1

1.1.2　Windows 95 安装 ……………………………………… 2

1.2　Windows 95 的启动及桌面环境 ……………………………… 3

1.2.1　Windows 95 的启动 …………………………………… 3

1.2.2　Windows 95 的界面环境 ……………………………… 3

1.3　基本操作 ………………………………………………………… 7

1.3.1　窗口的打开和关闭 ……………………………………… 7

1.3.2　最大化和最小化窗口 …………………………………… 9

1.3.3　窗口大小的改变、移动及安排 ………………………… 9

1.3.4　图标的排列和显示方式的改变 ………………………… 12

1.4　程序的运行和安装 ……………………………………………… 14

1.4.1　Start 菜单的使用 ……………………………………… 14

1.4.2　程序的运行 ……………………………………………… 15

1.4.3　程序的安装 ……………………………………………… 19

1.5　文件和文件夹的管理 …………………………………………… 22

1.5.1　两个管理工具 …………………………………………… 22

1.5.2　文件夹的操作 …………………………………………… 24

1.5.3　文件操作 ………………………………………………… 26

1.6　Windows 95 的帮助系统及查找功能 ………………………… 29

1.6.1　帮助窗口的基本使用方法 ……………………………… 29

1.6.2　其他的帮助形式 ………………………………………… 34

1.6.3　查找功能 ………………………………………………… 38

1.7　退出 Windows 95 操作系统 …………………………………… 42

第二章　Visual Basic 4.0 简介 ………………………………… 44

2.1　Visual Basic 发展简史 ………………………………………… 44

2.2　Visual Basic 4.0 中新增加的内容 …………………………… 44

2.3　Visual Basic 用到的几个基本术语 …………………………… 46

2.4　Visual Basic 4.0 的安装 ……………………………………… 47

2.5　Visual Basic 4.0 的不同版本 ………………………………… 52

2.6　用 Visual Basic 建立 32 位应用程序 ………………………… 52

2.7　怎样使用 Visual Basic 的帮助系统 …………………………… 53

第三章　Visual Basic 4.0 编程环境 …… 57
3.1　启动 Visual Basic 4.0 …… 57
3.2　Visual Basic 的用户环境 …… 58
3.2.1　Visual Basic 各个窗口的作用 …… 58
3.2.2　工具栏 …… 68
3.2.3　菜单 …… 69
3.2.4　标题条(Title Bar) …… 80
第四章　建立一个简单的应用程序 …… 81
4.1　简单应用程序的设计 …… 81
4.1.1　建立用户界面的对象 …… 81
4.1.2　为对象设置属性值 …… 82
4.1.3　把程序代码加到对象的控件中去 …… 83
4.2　掌握运行模式的概念 …… 85
4.2.1　解释运行模式 …… 85
4.2.2　编译运行模式 …… 86
4.3　运行程序 …… 86
4.4　保存文件 …… 87
4.5　编译应用程序 …… 89
第五章　窗体和工具 …… 90
5.1　基本概念 …… 90
5.1.1　窗体 …… 90
5.1.2　控件 …… 90
5.1.3　属性 …… 90
5.2　定制窗体 …… 91
5.2.1　生成窗体 …… 91
5.2.2　窗体的几种快捷操作 …… 93
5.2.3　属性<Properties>窗口 …… 93
5.2.4　窗体的属性 …… 98
5.2.5　窗体的相关事件 …… 102
5.2.6　窗体对事件的响应 …… 103
5.3　控件操作 …… 107
5.3.1　控件的生成 …… 107
5.3.2　同时操作多个控件 …… 108
5.3.3　控件的删除 …… 110
5.4　控件的属性及相关事件 …… 110
5.4.1　标签 …… 110
5.4.2　文本框 …… 110
5.4.3　命令按钮 …… 112
5.4.4　框架 …… 112

5.4.5 单选按钮 …… 112
5.4.6 复选框 …… 113
5.4.7 列表框 …… 113
5.4.8 组合框 …… 114
5.4.9 水平和垂直滚动条 …… 114
5.5 控件对事件的响应 …… 115
5.6 存储用户的工作 …… 117
5.6.1 存储一个新的工程文件 …… 117
5.6.2 用新的文件名存储已存在的工程文件 …… 118
5.6.3 存储新的窗体，标准或类模块 …… 118
5.6.4 用新的文件名存储窗体或类模块文件 …… 118
5.6.5 自动存储用户工作 …… 118
5.7 建立 Windows 应用程序 …… 119
5.8 多重窗体 …… 121
第六章 基本语言 …… 124
6.1 变量 …… 124
6.1.1 声明变量 …… 124
6.1.2 变量类型 …… 124
6.1.3 变量说明 …… 126
6.1.4 全局变量与局部变量 …… 127
6.2 基本语句 …… 128
6.2.1 条件语句 …… 128
6.2.2 循环语句 …… 130
6.2.3 转移语句 …… 131
6.3 常用函数 …… 132
6.3.1 字符串函数 …… 132
6.3.2 转换函数 …… 134
6.3.3 数学函数 …… 135
6.3.4 日期函数 …… 136
6.4 数组的使用 …… 138
6.5 数字表达式的计算 …… 139
6.5.1 算术运算符 …… 139
6.5.2 关系运算符 …… 139
6.5.3 逻辑运算符 …… 139
6.6 赋值语句和属性设置 …… 140
6.7 修改 myfirst 工程文件 …… 141
第七章 菜单设计 …… 143
7.1 Menu Editor（菜单编辑器）窗口简介 …… 143
7.2 建立菜单 …… 146

7.2.1 建立菜单条 …… 147
7.2.2 把代码连接到菜单项上 …… 152
7.3 试验菜单命令 …… 154
7.4 上托菜单 …… 156
第八章 对话框 …… 158
8.1 创建对话框的基本知识 …… 158
8.1.1 怎样定制一个对话框的界面风格 …… 158
8.1.2 调入和显示对话框 …… 158
8.1.3 为对话框的控件设置属性 …… 159
8.2 预制对话框 …… 159
8.2.1 消息框 …… 159
8.2.2 输入框 …… 164
8.3 控件的建立 …… 167
8.3.1 单选按钮的建立 …… 167
8.3.2 复选框的建立 …… 168
8.3.3 文本框的建立 …… 172
8.3.4 列表框的建立 …… 175
8.3.5 组合框的建立 …… 179
8.4 定制对话框 …… 182
8.5 公用对话框 …… 183
8.5.1 Open 对话框 …… 185
8.5.2 Save As 对话框 …… 187
8.5.3 Color 对话框 …… 188
8.5.4 Print 对话框 …… 189
8.6 打印操作 …… 189
8.6.1 与打印操作有关的方法和属性 …… 189
8.6.2 直接将信息送往打印机 …… 192
8.6.3 窗体输出 …… 193
第九章 图形及绘图操作 …… 194
9.1 图像工具 …… 194
9.1.1 Picture Box(图片框)工具 …… 194
9.1.2 Image(图像框)工具 …… 199
9.1.3 保存图像内容 …… 203
9.1.4 图片框与图像框的区别 …… 205
9.2 绘图工具 …… 206
9.2.1 Line(直线)工具 …… 206
9.2.2 用 Line 方法绘图 …… 208
9.2.3 Shape(形状)工具 …… 213
9.2.4 用 Circle 方法绘图 …… 216

9.3　颜色的使用 …… 220
9.4　定制标度 …… 222
9.5　PSet 方法绘图 …… 222
9.6　绘制曲线 …… 225
9.7　简单动画设计 …… 227
　9.7.1　制定 Form 坐标系 …… 228
　9.7.2　在预定义坐标系中移动对象 …… 228
　9.7.3　用 Move 方法和时钟(Timer)控件建立动画 …… 228
　9.7.4　动画设计 …… 228
9.8　在 Form 窗口中添加背景图形 …… 232
第十章　文件操作 …… 235
10.1　文件系统概述 …… 235
10.2　文件系统控件 …… 235
　10.2.1　文件列表框 …… 236
　10.2.2　目录列表框 …… 237
　10.2.3　驱动器列表框 …… 238
　10.2.4　协同文件系统控件 …… 239
10.3　文件基本操作 …… 241
　10.3.1　准备工作 …… 241
　10.3.2　文件的删除 …… 243
　10.3.3　拷贝和移动文件 …… 244
　10.3.4　文件的更名 …… 247
　10.3.5　运行程序 …… 249
10.4　顺序文件 …… 251
　10.4.1　打开和关闭顺序文件 …… 251
　10.4.2　读和写顺序文件 …… 252
　10.4.3　一个编辑 Autoexec.bat 的应用程序 …… 256
10.5　随机文件 …… 259
　10.5.1　打开和关闭随机文件 …… 260
　10.5.2　读和写随机文件 …… 260
　10.5.3　一个学生档案数据库程序 …… 261
10.6　二进制文件 …… 269
第十一章　错误处理与程序调试 …… 270
11.1　错误类型 …… 270
　11.1.1　语法错误 …… 270
　11.1.2　编译错误 …… 272
　11.1.3　运行时发生的错误 …… 272
　11.1.4　逻辑错误 …… 273
11.2　错误处理 …… 274

11.2.1　使用错误处理程序…… 275
11.2.2　设置错误陷阱(trap):使用 On Error 语句 …… 275
11.2.3　使用 On Error GoTo 语句…… 277
11.2.4　使用 Error 和 Error $ 函数 …… 280
11.2.5　使用 Error 语句 …… 282
11.2.6　使用 Err. Description 属性…… 284
11.3　使用 Debug(调试)窗口 …… 285
11.4　设置断点…… 293
第十二章　Visual Basic 4.0 高级技术 …… 295
12.1　剪贴板的应用…… 295
12.2　数据访问…… 296
12.2.1　数据控件…… 296
12.2.2　显示数据库信息…… 296
12.2.3　寻找记录…… 300
12.2.4　增添记录…… 303
12.2.5　删除记录…… 305
12.3　数据管理器…… 307
12.3.1　创建新的数据库…… 308
12.3.2　为数据库添加信息…… 313
12.3.3　表操作…… 316
12.4　对象的链接与嵌入(OLE)…… 318
12.4.1　一些 OLE 术语 …… 318
12.4.2　对象的链接与嵌入…… 319
12.4.3　OLE 自动化(Automation) …… 325
12.5　定制控件概述…… 330
12.5.1　定制控件对话框…… 330
12.5.2　DBList 和 DBCombo 定制控件的应用 …… 332
12.5.3　可插入对象的基本使用方法…… 334
附录　Visual Basic 4.0 术语 …… 338

前　言

Microsoft Visual Basic 是 BASIC 语言以及所有计算机语言发展过程中的一个重要的分支。用传统的 BASIC 解释程序或编译程序编制的程序生成GUI（图形用户界面）应用程序的过程既冗长又乏味，用户为了在屏幕上生成和绘制窗口，必须编制大量的代码，必须亲自绘制窗口中的按钮和别的控件，还要亲自跟踪键盘和鼠标输入。

Visual Basic 的重要特点之一就是可视。为此，引进了窗体和控件的概念和机制，用于设计应用程序界面。用户通过在屏幕上生成窗体，在窗体中绘制按钮和控件，设计菜单和对话框，Visual Basic 会替用户生成在屏幕上绘制上述图形的代码，并为用户跟踪鼠标和键盘的交互输入。用户根本就看不到这些代码，这由 Visual Basic 在后台进行处理。

在过去，一个大型应用软件的代码中，有高达 90%的部分是用来处理用户界面的，而只有 10%的代码是用来处理实际功能的。所以，切换到能处理大部分用户界面的编程环境中（仍然需要生成用户界面并绘制它），会使编程效率得到很大程度的改观。现在，在数十分钟内用户就能编制一个很专业化的 Windows 应用程序，并带有所有的窗口化界面、按钮、复选框以及图形窗口，而不是像过去那样花费大量的时间去考虑如何制作用户界面。把这些时间用于完成程序的功能部分，必将使工作效率得到很大程度的提高。

Visual Basic 的另一重要特点是事件驱动，当窗口或控制上发生一个事件（Event）时，Visual Basic 就把控制权交给程序员。例如，当用户单击鼠标一个按钮时，程序员不必担心如何判断按钮是否已被按动，或者编写代码显示按钮被按动的过程，只需要编写一段代码（事件处理例程）告诉计算机在按钮被按动后，需要做些什么工作就可以了。这样，设计一个 Windows（包括 Windows 3.1、Windows 95 和 Windows NT）应用程序就变成用鼠标在屏幕上点按、拖拉并添加少量代码就可以完成了，大大简化了程序设计的难度，降低了对程序设计的要求，加快了应用程序开发的速度。

Visual Basic 4.0 有两种版本，即 16 位和 32 位版本。16 位版本的 Visual Basic 4.0 支持 Windows 3.1 和 Windows 95 应用程序的开发，32 位版本的 Visual Basic 4.0 支持 Windows 95 和 Windows NT 应用程序的开发。不管是那个版本，OLE 和定制控件的功能都得到了显著的增强。

本书以循序渐进的内容编排，结合丰富的图例，使读者可以轻松地学习和掌握 Visual Basic 4.0。Windows 95 是一个崭新的操作系统，本书首先介绍 Windows 95 的特点和基础的操作方法，在此基础上，介绍 Visual Basic 的历史和 4.0 版的安装；接着介绍 Visual Basic 的编程环境，讨论如何建立一个简单的应用程序；并说明了窗体和控件的概念和设计方法；解说 Visual Basic 的基本语言；阐述菜单和对话框的设计、图形和绘图操作、文件处理以及错误处理与程序调试技术；最后介绍了 Visual Basic 4.0 的高级程序设计技术，包括 OLE、定制控件和数据库管理等。在附录中，还给出了 Visual Basic 4.0 的术语。

作者虽尽力认真工作，但由于水平有限，加上成书较快，书中错误难免，欢迎读者指正。

编　者
1995 年 12 月

第一章　Windows 95 入门

1.1　Microsoft Windows 95 的概述及安装

1.1.1　Windows 95 概述

Microsoft 终于推出了 Windows 95，对用户来说这是一件令人兴奋不已的事。因为 Windows 95 将为用户提供更方便的操作、更为友好的界面和网络功能。

利用 Windows 95 操作系统，用户现在所做的工作将变得更加简便和快捷，多年的期待——从电子邮件到 Internet 联网，现在都将成为现实。Windows 95 使计算机变得更为友好、更易于操作。因此无论做什么，都会享受到无穷的乐趣。用户首先看到的是重新设计过的图形界面，它不仅易于使用，而且效率更高，具有更强的自定义功能。另外，在 Windows 95 中安装和配置兼容“即插即用”的调制解调器、CD-ROM 驱动器以及其他的外围设备，已不仅仅是简便易行，全新的“即插即用”技术可以使安装和配置工作自动完成。用户使用 Windows 95 还会有新的收获，Windows 95 为便携机用户提供了增强的多媒体技术和更强大的计算功能。Windows 95 还提供了集成的、无缝的网络技术，并支持“infobahn”漫游。

由于 Windows 95 是为运行新一代的 32 位程序而开发的，因此用户可以充分利用其强大的新增特性，如抢先式多任务和多线程功能，使用户可同时执行多个操作。此外，Windows 95 为用户当前基于 MS-DOS 和 Windows 的程序提供了一个更稳定的运行环境。在娱乐方面，Windows 95 提供了更强大更稳定的显示技术，并且增强了支持快节奏游戏的能力。更便捷更有效的工作方式会使用户抛开每天的烦恼，舒心地投入工作。Windows 95 提供了下列更便捷、更迅速的工作方式：

（1）Start 按钮始终显示在屏幕上。在任何时候单击此按钮，都可以使用 Windows 95 在打开程序、文件管理、系统维护以及其他方面的强大功能。

（2）利用任务栏，很容易在多个程序之间进行切换。每当用户启动程序时，一个代表该程序的按钮就会出现在任务栏上。如果要使用某一个程序只要单击该程序的按钮即可。

（3）“即插即用”使用户不必再以人工方式设置硬件设备。Windows 95 可以自动检测“即插即用”兼容设备并对其进行设置。

（4）Windows 95 能够与现有硬件很好地配合，因此用户可最大程度地利用现有的为 MS-DOS 和 Windows 操作系统所写的应用程序。

（5）利用快速磁盘和文件存取以及快速打印功能，用户可大大提高工作效率。

（6）可以更有效地使用资源，只要单击鼠标右键，就会显示一个包含对当前对象进行操作的常用命令的菜单，使操作更加方便。

（7）由于具备支持 NetWare、Windows NT Server 网络操作系统以及 Windows for Workgroups 的内置客户机，网络安装过程得以简化。此外，Windows 95 还支持所有主要的网络传输驱动程序标准。

开放的环境提供了更广阔的天地：

(1) 同时完成多任务。通过运行 32 位程序，Windows 95 提供了抢先式多任务和多线程机制，并且使程序在它们各自的内存空间中运行。

(2) 利用内置的 Microsoft Exchange 收件箱，用户可以发送和接收电子邮件及传真。

(3) Windows 95 内置了对连接 Internet 的支持，它采用集成访问并在操作系统中内置所有必要的组件。可以在任何地方与用户所在机构的网络相连。内置的拨号联网功能降低了远程计算的复杂性，因此无论用户身在何处，联网都将是一件轻松的事。

(4) 使用便携机更加方便。Windows 95 为便携机提供了更丰富的实用功能，包括在两台 PC 间更新文件的 Briefcase，对 PC 卡和锚站及 Battery Meter 提供了即插即用支持。

寓工作于娱乐：

(1) Windows 95 改进的界面使用户在使用系统时感觉更方便，常用功能的使用都有 Wizard 提供帮助，Help 系统的功能也更加强大。

(2) 更快、更平稳的视频效果，使 Windows 95 在多媒体和游戏方面显示出卓越的性能。

(3) 利用 Autoplay，各种多媒体产品更容易使用。只要插入可利用自动播放功能的 CD，Windows 95 就可以自动进行播放。

1.1.2 Windows 95 安装

Windows 95 的安装同 Windows 3.1 基本上是一样的。整个操作系统的软件一般存放在 25 张 3.5 英寸的软盘上。如果用户拥有光盘驱动器，也可以用一张光盘进行安装。本书以当前主流机 386 和 486 为例。在安装前，用户得先确定一下系统的硬件环境是否满足操作系统的需要。Windows 95 需要以下系统环境：

最低配置： Intel 386 CPU
60MB 可用硬盘空间
4MB 内存
640×480 VGA 显示器

最优配置： Intel 486 或 Pentium CPU
200MB 以上硬盘
8MB 或 16MB 内存
TVGA 或 SVGA 显示器

如果用户确定自己的系统具有最低配置以上的硬件环境，并具有 3.5 英寸的软驱或光盘驱动器，那么就可以安装 Windows 95 了。具体步骤如下：

一、启动 DOS 系统，在 3.5 英寸的软驱中插入 1 号磁盘，然后在 DOS 提示符下键入 b:\setup 并回车，这时屏幕将显示操作信息，表示系统正在查看内存。如果用户机为膝上型计算机，则安装程序将提示用户去掉常驻内存中的 POWER.EXE 程序，用户这时需要退出安装程序(按 ESC 键)，重新编辑 C:\CONFIG.SYS 文件，去掉 POWER.EXE 文件，存盘后重新启动，再运行安装程序。台式计算机则不会出现该问题，不过，若用户机内存的最低配置仅为 4MB 内存，而又运行了磁盘缓冲程序 SMARTDRV.EXE 的话，则 Setup 将提示内存不够。这需要在 AUTOEXEC.BAT 文件中删除该命令。

二、安装程序接着运行磁盘扫描程序 Scandisk 检查 C 盘的状态。如果 C 盘的空间足够，而且没有异常，则安装程序将提示系统检测通过，可以准备安装，并建议用户在 Windows 95

操作系统安装以后，定期运行 Scandisk 以检测磁盘的错误并修改发现的错误。如果硬盘存在问题，Scandisk 修正错误，然后提示用户：

(1) 继续安装程序，按 Enter 键。

(2) 退出安装程序，按 ESC 键。

用户可以按 Enter 键继续安装过程。

三、这时，安装程序初始化你的系统，如果用户的系统中原来存有某一低版本 Windows 操作系统，则安装程序应从 Windows 操作系统下运行。这样，将加快系统的安装速度，因为这种系统的安装实际上就是系统版本的升级，而且可以节省硬盘空间。如果用户不想毁坏原来的系统，比如中文 Windows 3.1 操作系统，则可直接在 MS-DOS 下安装。

(1) 如果想退出安装，改到 Windows 操作系统下运行安装程序，则按 Enter 键。

(2) 如果想继续在 MS-DOS 下运行安装程序，则按 ESC 键。

四、这时屏幕上出现了 Windows 95 Setup 欢迎使用画面，并提示用户 Windows 95 安装过程需要 30 到 60 分钟的时间(依据系统性能高低)，单击 Continue。

五、这时屏幕出现一信息框，表示安装程序开始准备从安装盘上装入安装指南到内存以指导用户后面的安装工作，请用户稍候。准备工作结束以后，屏幕弹出一对话框，提示用户插入 2 号盘。

六、取出 1 号盘，插入 2 号盘，单击 OK 或按回车。接下来用户可以依据屏幕上提供的信息一步一步地完成 Windows 95 的安装工作。

在用户安装 Windows 95 的过程中，安装程序会提醒用户选择适当的安装类型，根据用户的需要，可以选择以下安装类型：

(1) Typical：该选项用于当前主流机的配置，在多数台式计算机上可以选择该项。

(2) Portable：该选项将 Windows 95 设置为适用于便携式计算机和运动型计算机。

(3) Compact：该选项用于紧凑型配置，即适用于那些硬盘空间紧张的用户计算机。为了给系统节省磁盘空间，Setup 将不为用户安装可有可无的元部件，即为用户安装一个最基本的 Windows 95。

(4) Custom：该选项仅用于高级用户和系统管理者，该选项将为用户提供所有可得的 Setup 选项，以便用户按需要定制元部件。

1.2　Windows 95 的启动及桌面环境

1.2.1　Windows 95 的启动

Windows 95 启动很方便，就像 MS-DOS 启动一样，开机便能够启动 Windows 95。

1.2.2　Windows 95 的界面环境

开机运行 Windows 95 后，屏幕上出现带有各种图标的非常友好的环境，它称为桌面环境。见图 1.1。其中屏幕底部是任务栏，它包括一个 Start 按钮。Start 按钮能够用来快速地运行程序或发现文档，它也是得到帮助的最快方法。当打开一个程序或文档时，一个相应的按钮出现在任务栏上，用这些按钮可以很快在打开的程序之间进行切换。

因为计算机的配置的不同，桌面环境上出现的图标也略有不同，一般包括以下几项：

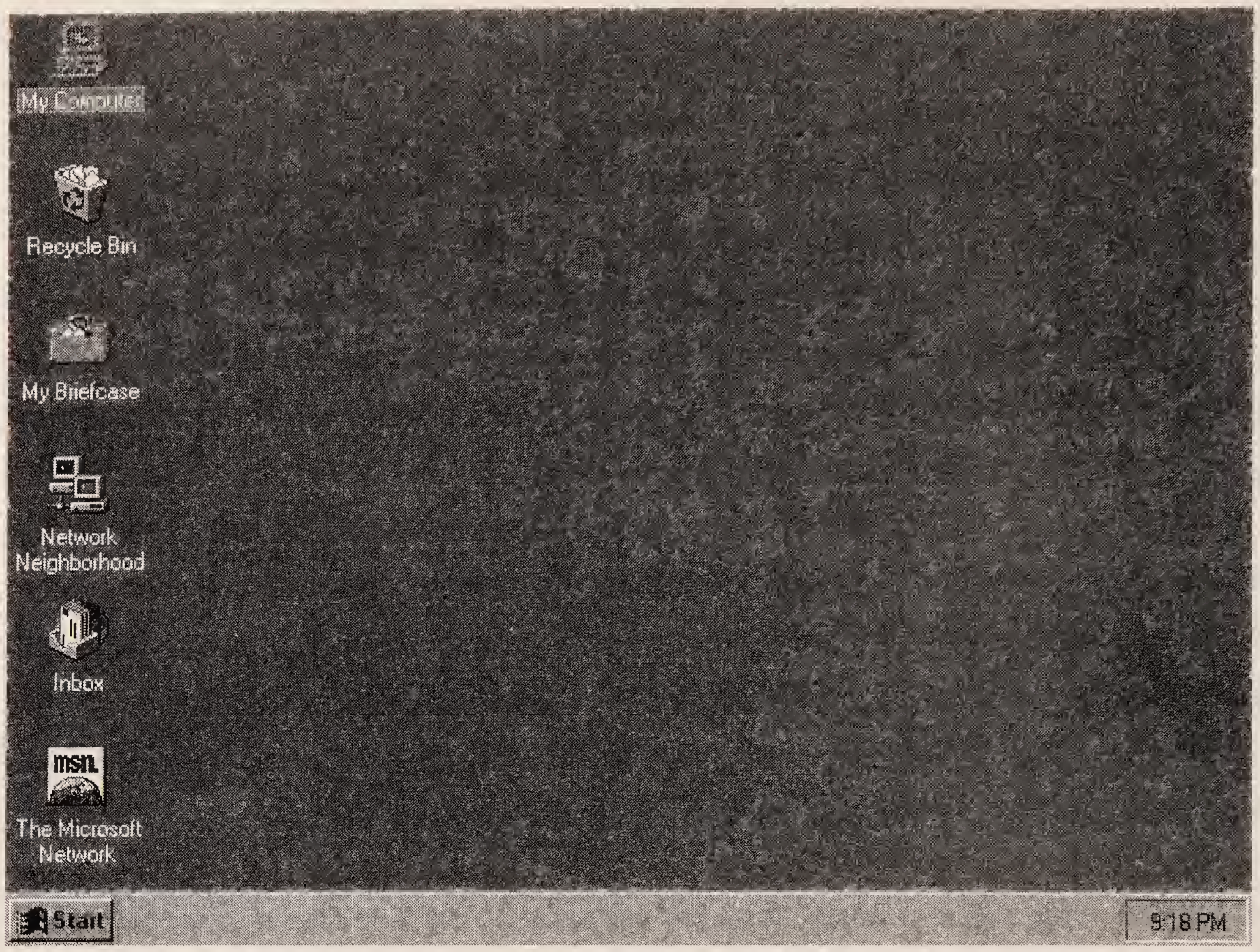

My Computer
Recycle Bin
My Briefcase
Network Neighborhood
Inbox
The Microsoft Network
Start
9:18 PM

图 1.1 桌面环境

● **My Computer（我的电脑）**：用它可以非常方便地查看电脑中的所有资源。双击 My Computer 图标，出现图 1.2 的画面。如果双击“Ms-DOS_6(C:)”图标，就可以浏览 C：硬盘上的所有目录和文件，见图 1.3。

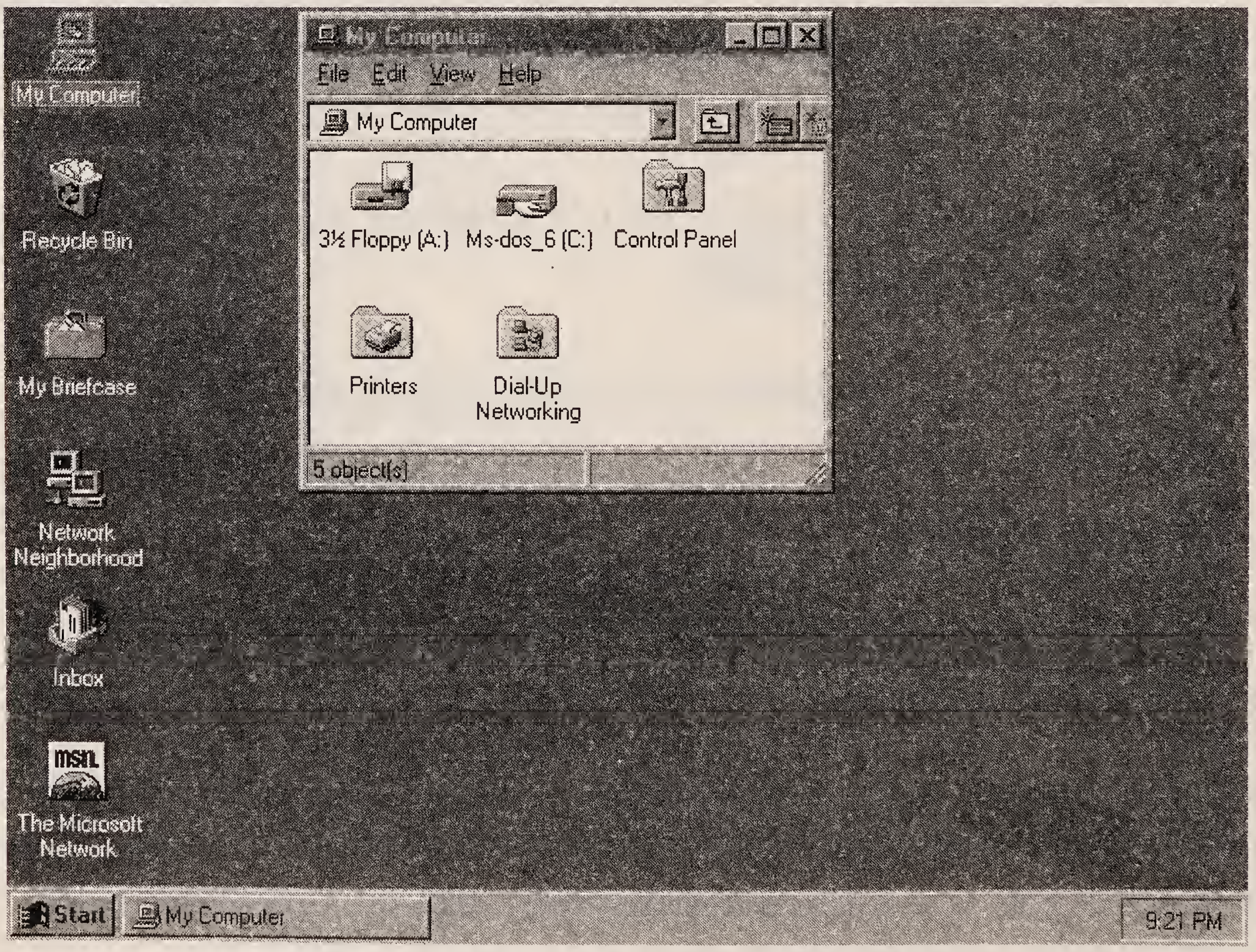

My Computer
Recycle Bin
My Briefcase
Network Neighborhood
Inbox
The Microsoft Network
My Computer
File
Edit
View
Help
My Computer
3½ Floppy (A:)
Ms-dos_6 (C:)
Control Panel
Printers
Dial-Up Networking
5 object(s)
Start
My Computer
9:21 PM

图 1.2 My Computer

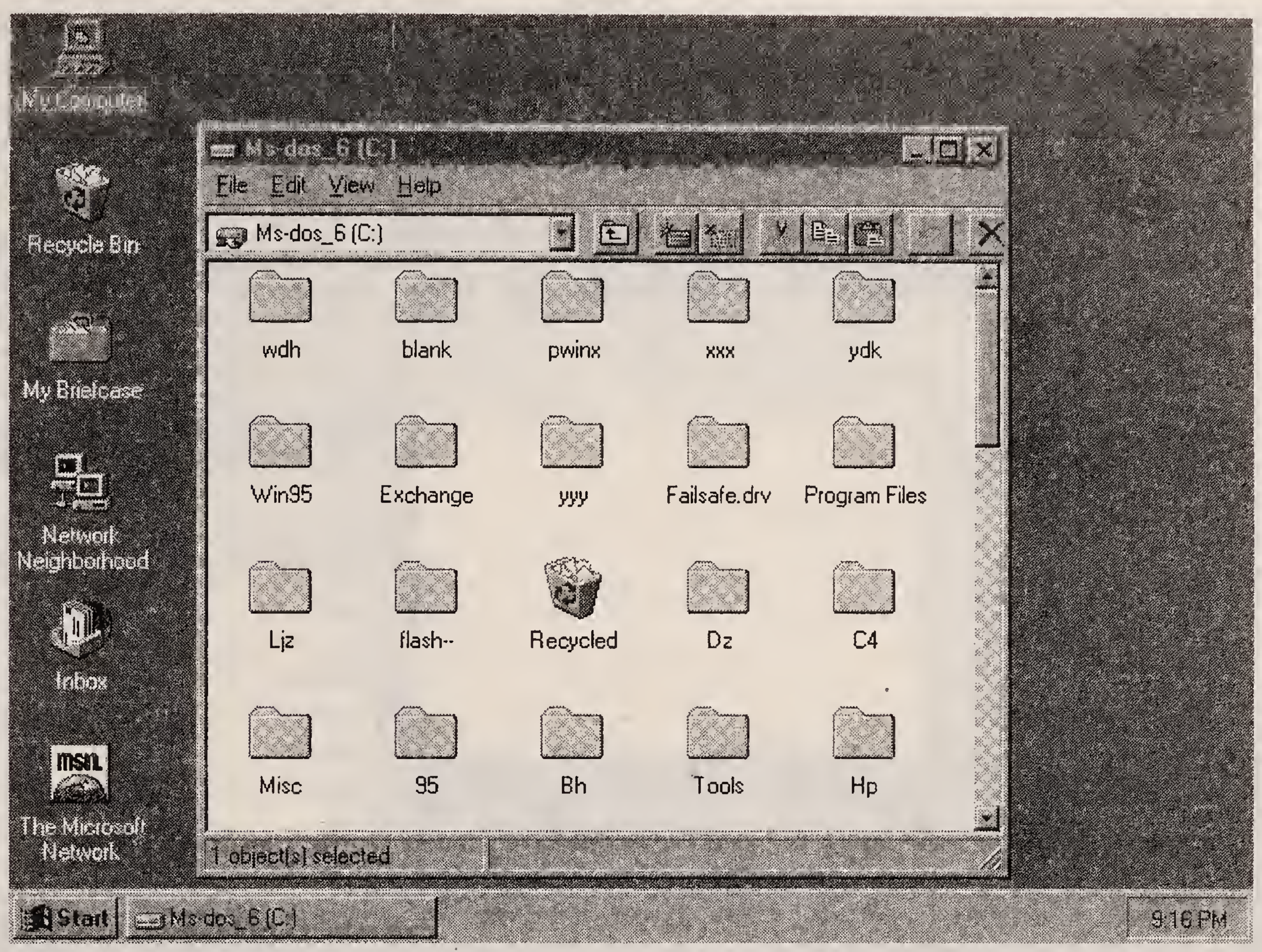

My Computer
Recycle Bin
My Briefcase
Network Neighborhood
Inbox
The Microsoft Network
Ms-dos_6 (C:)
File Edit View Help
Ms-dos_6 (C:)
wdh
blank
pwinx
xxx
ydk
Win95
Exchange
yyy
Failsafe.drv
Program Files
Ljz
flash--
Recycled
Dz
C4
Misc
95
Bh
Tools
Hp
1 object(s) selected
Start
Ms-dos_6 (C:)
9:16 PM

图 1.3　浏览硬盘资源

● **Recycle Bin（回收站）**：是删除文档的暂时存放处。当删除一个文档时，它并没有从硬盘上真正消失，而是被存放于回收站中，以备恢复被删除的文档时使用。双击回收站图标就可打开它的窗口，见图 1.4。窗口中列出被删除的文档，单击 File 菜单的 Empty Recycle Bin 选项和 Restore 命令分别执行清空（从硬盘上真正删除）和恢复操作，分别见图 1.5 和图 1.6。

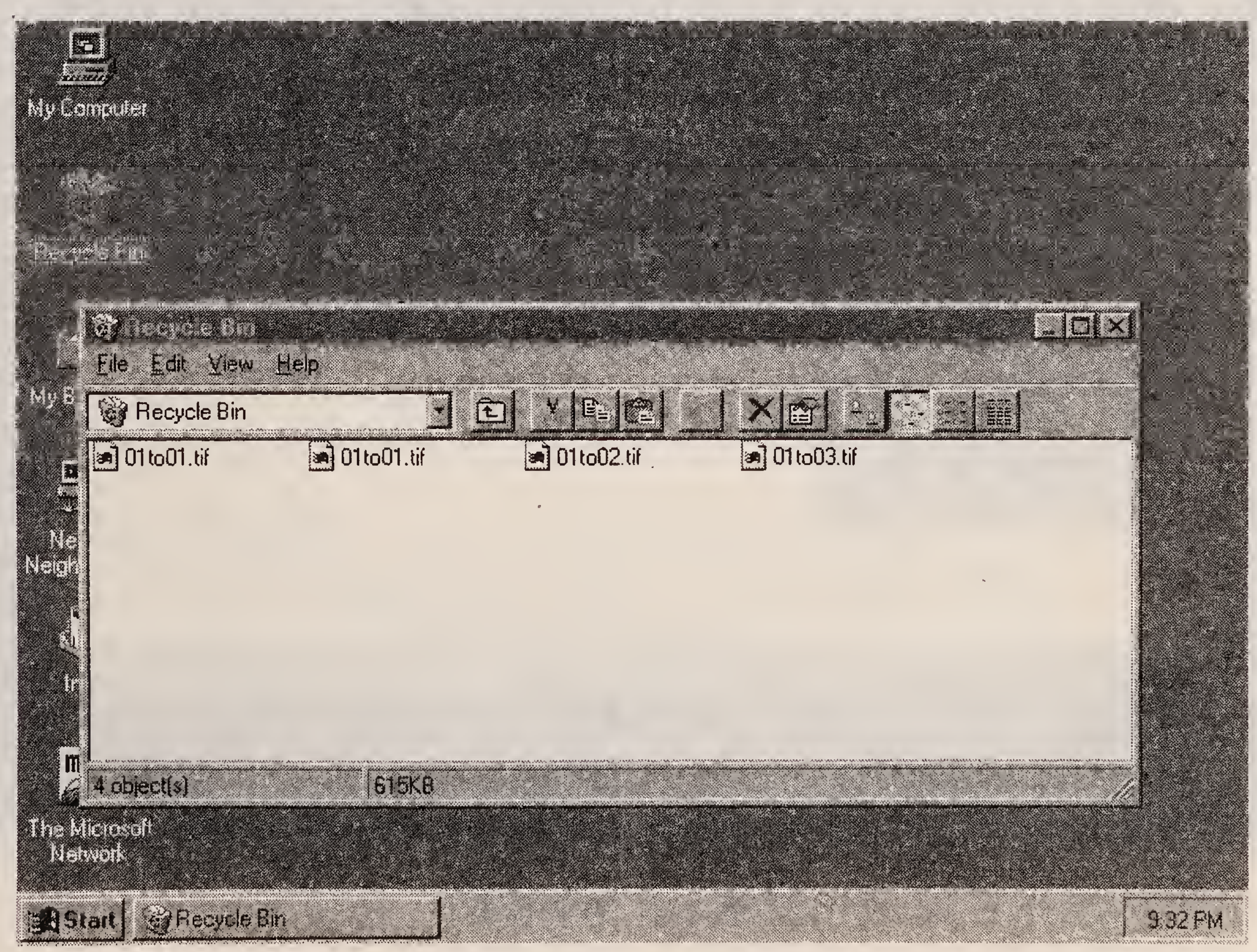
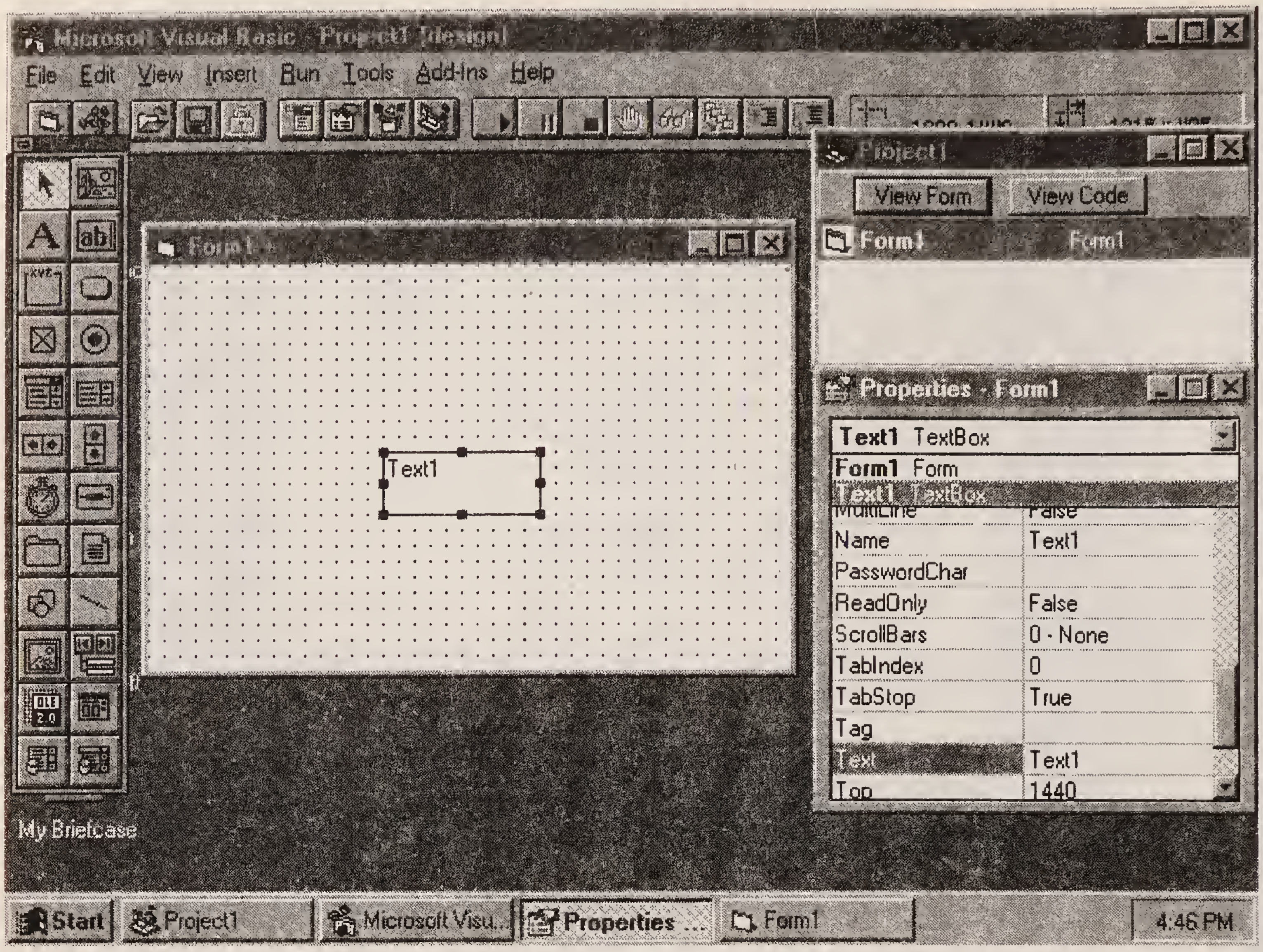
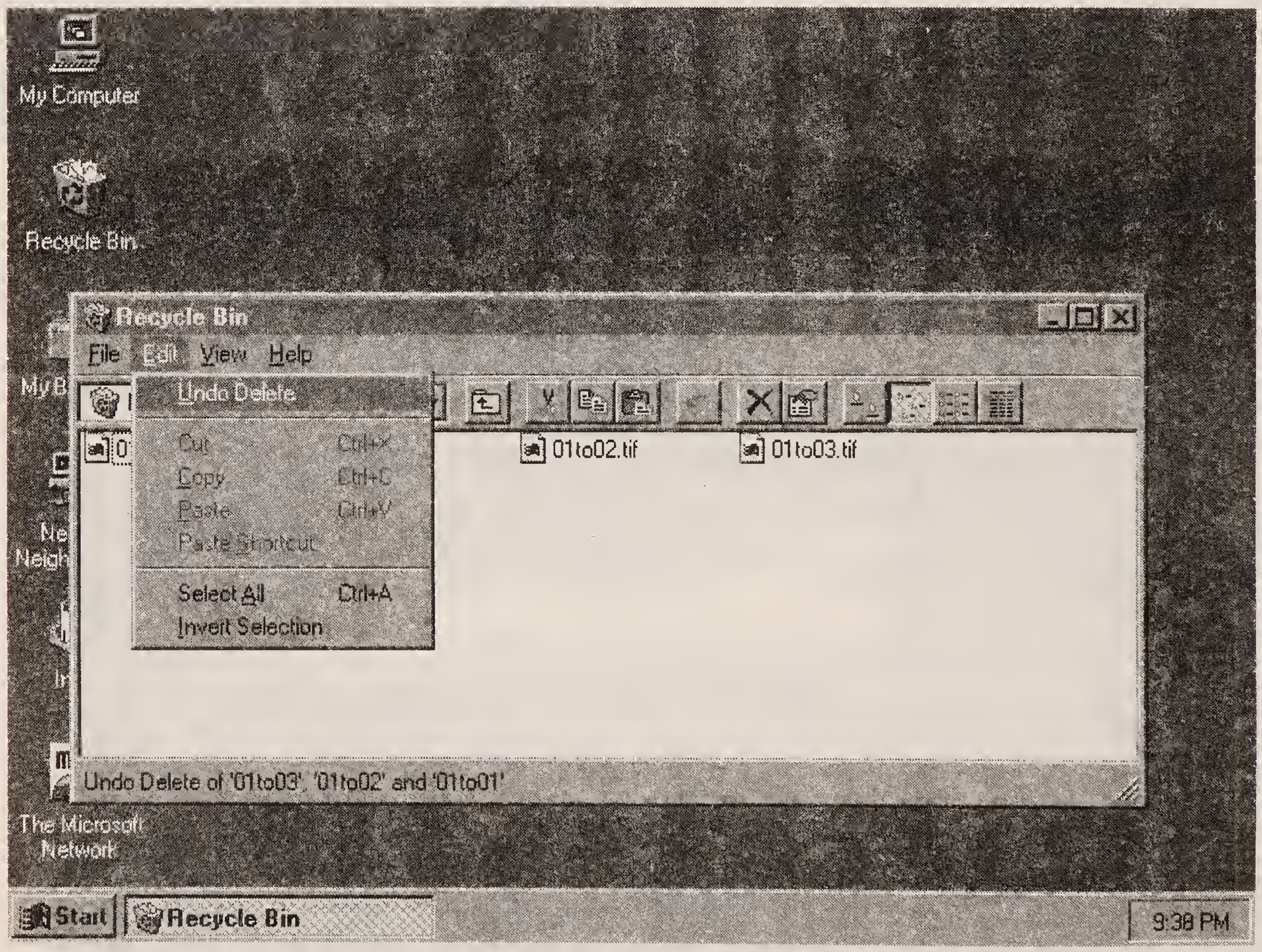

My Computer
Recycle Bin
The Microsoft Network
Recycle Bin
File Edit View Help
Recycle Bin
01to01.tif
01to01.tif
01to02.tif
01to03.tif
4 object(s)
615KB
Start
Recycle Bin
9:32 PM

图 1.4　回收站


Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
Form1
Text1
Project1
View Form
View Code
Form1 Form1
Properties - Form1
Text1 TextBox
Form1 Form
Text1 TextBox
My Briefcase
Start
Project1
Microsoft Visu...
Properties ...
Form1
4:46 PM

MultiLine	False
Name	Text1
PasswordChar	
ReadOnly	False
ScrollBars	0 - None
TabIndex	0
TabStop	True
Tag	
Text	Text1
Top	1440

图 1.5 清空命令


My Computer
Recycle Bin
Recycle Bin
File Edit View Help
Undo Delete
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Paste Shortcut
Select All Ctrl+A
Invert Selection
01to02.tif
01to03.tif
The Microsoft Network
Start
Recycle Bin
9:38 PM

Undo Delete of '01to03', '01to02' and '01to01'

图 1.6 恢复命令

● **Network Neighborhood（网上邻居）**：如果用户的计算机已和其他机器连网，双击此图标，可看见网络中资源，见图 1.7。由图中可以看出，用户网络中有一台便携机和一台 486dx-100。

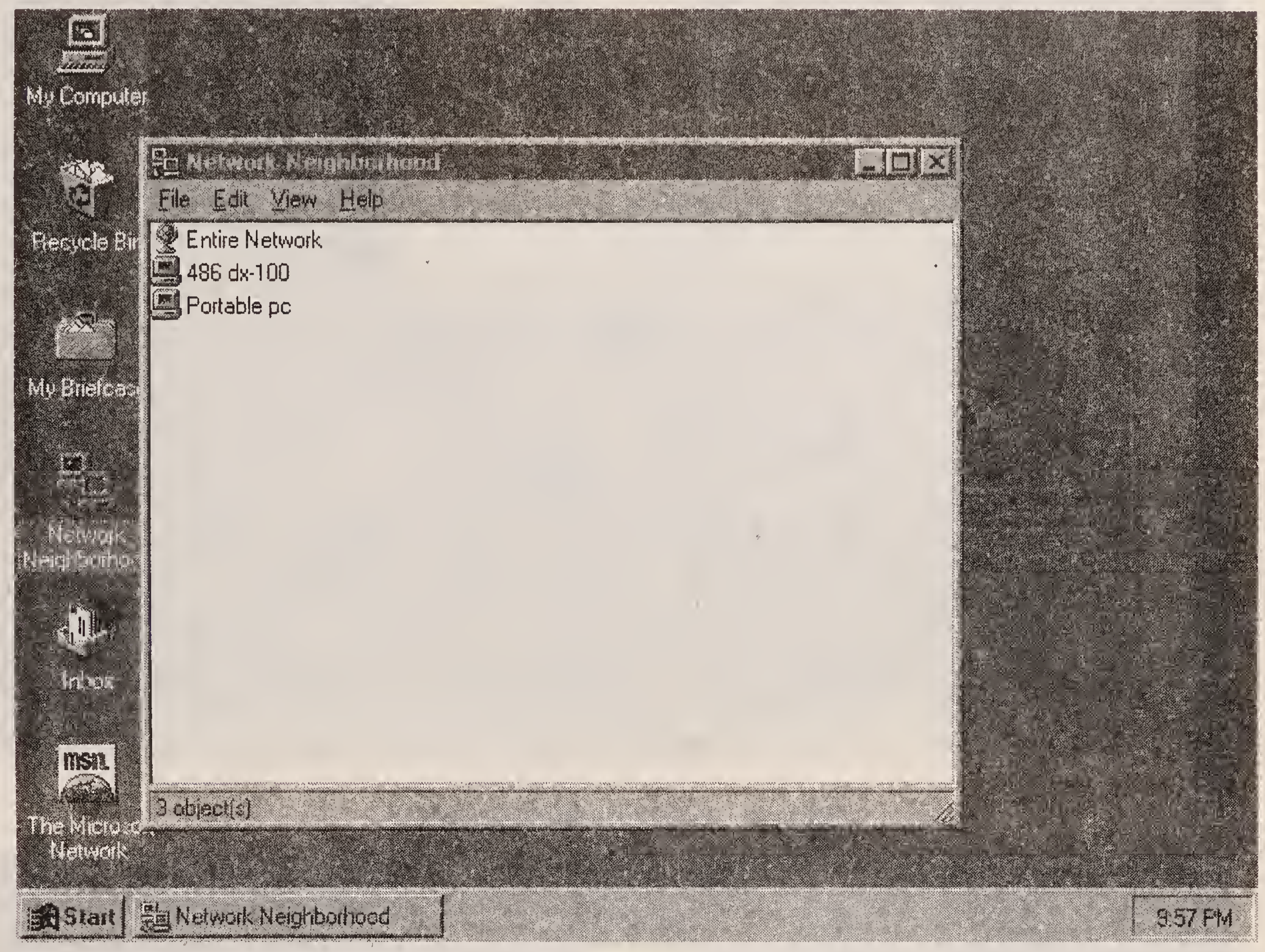

My Computer
Recycle Bin
My Briefcase
Network Neighborhood
Inbox
msn
The Microsoft Network
Network Neighborhood
File　Edit　View　Help
Entire Network
486 dx-100
Portable pc
3 object(s)
Start
Network Neighborhood
9:57 PM

图 1.7　网络中资源

● **Inbox（收件箱）**：主要用于通信中存放收发信件。

1.3　基 本 操 作

1.3.1　窗口的打开和关闭

其实前面已涉及到了打开一个窗口的方法，只要双击图标即可，同时屏幕底部任务栏上出现相应的按钮（见图 1.2）。

关闭一个窗口的方法有多种：

（1）在图 1.2 中，单击 File 菜单，再单击菜单下的 Close 命令，见图 1.8。

（2）在图 1.2 中，单击窗口右上端带“X”号的图标。

（3）在图 1.2 中，单击窗口左上端的控制按钮，此时出现一个菜单，单击 Close 命令，见图 1.9。或者直接双击控制按钮。

（4）在键盘上同时按下 Alt 和 F4 键。

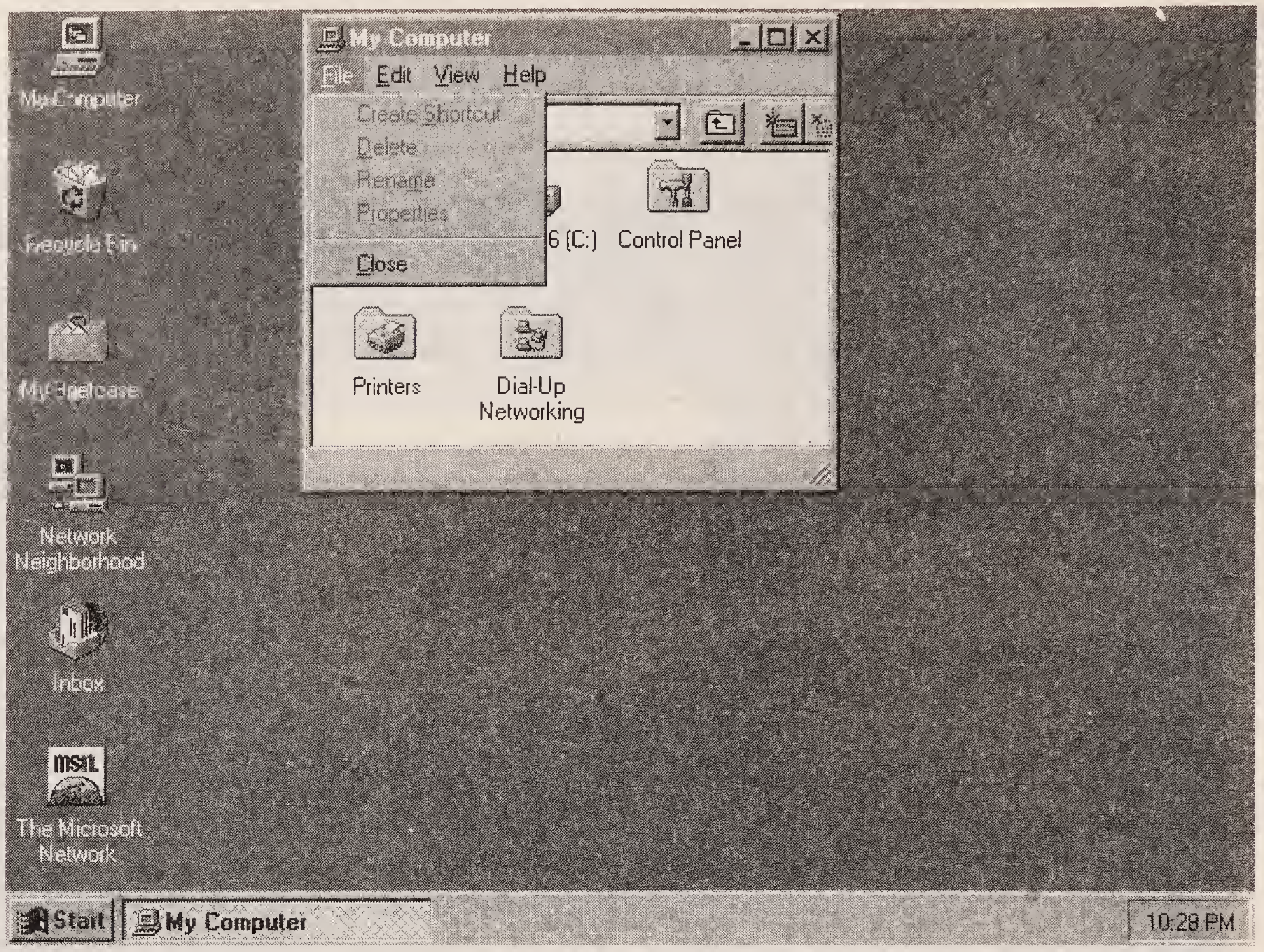
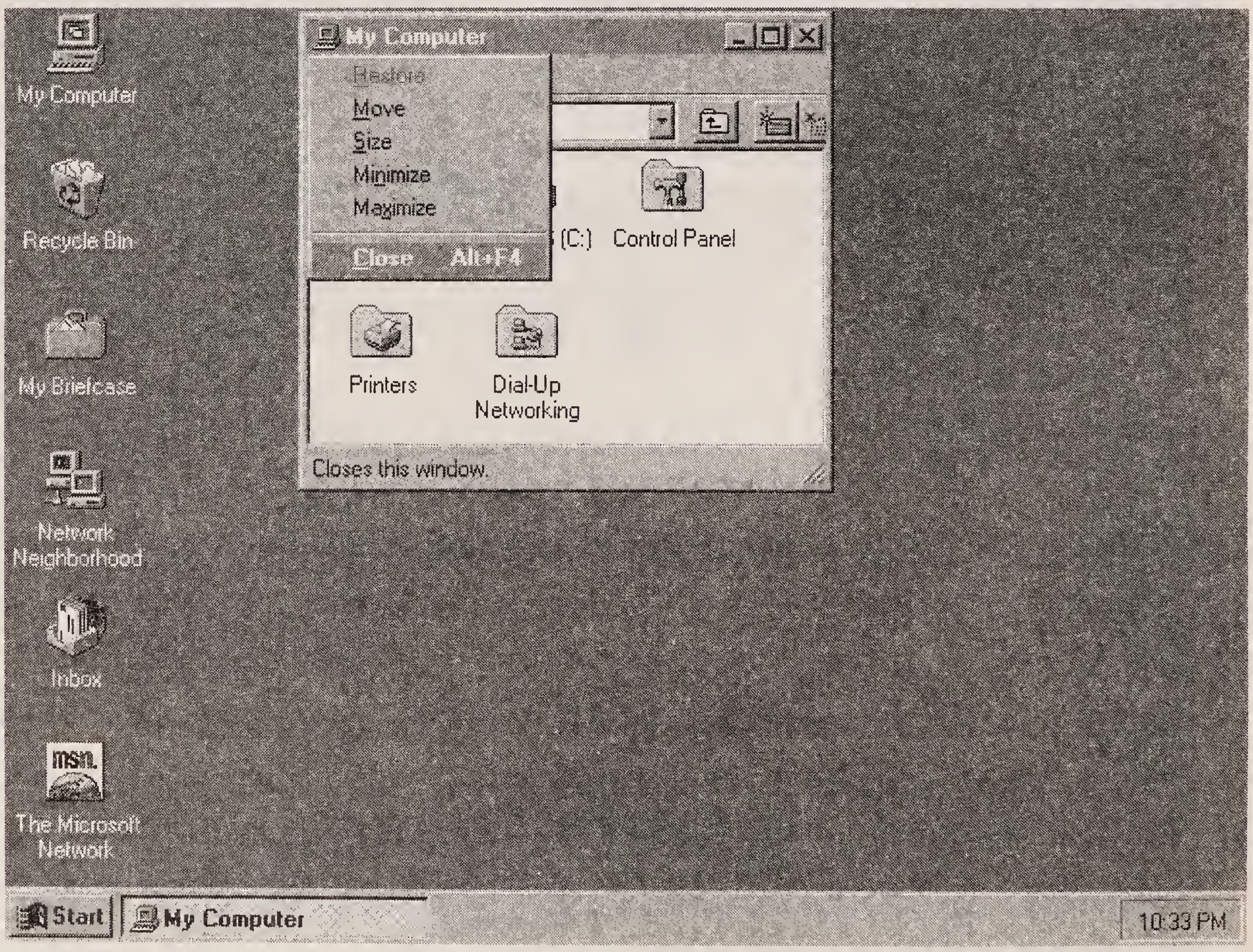

My Computer
File
Edit
View
Help
Create Shortcut
Delete
Rename
Properties
Close
6 (C:)
Control Panel
Printers
Dial-Up Networking
My Computer
Recycle Bin
My Briefcase
Network Neighborhood
Inbox
The Microsoft Network
Start
My Computer
10:28 PM

图 1.8 利用 File 菜单关闭窗口


My Computer
Restore
Move
Size
Minimize
Maximize
Close Alt+F4
(C:)
Control Panel
Printers
Dial-Up Networking
My Computer
Recycle Bin
My Briefcase
Network Neighborhood
Inbox
The Microsoft Network
Start
My Computer
10:33 PM

Closes this window.

图 1.9 利用控制菜单关闭窗口

1.3.2 最大化和最小化窗口

（1）最大化窗口

在图 1.2 中的窗口右上端有三个小图标，单击第二个带有小方格的图标就可以最大化窗口，此时屏幕变成图 1.10。在图中可以看到第二个图标变成重叠的双方格，单击此处就可以恢复窗口到原来大小。

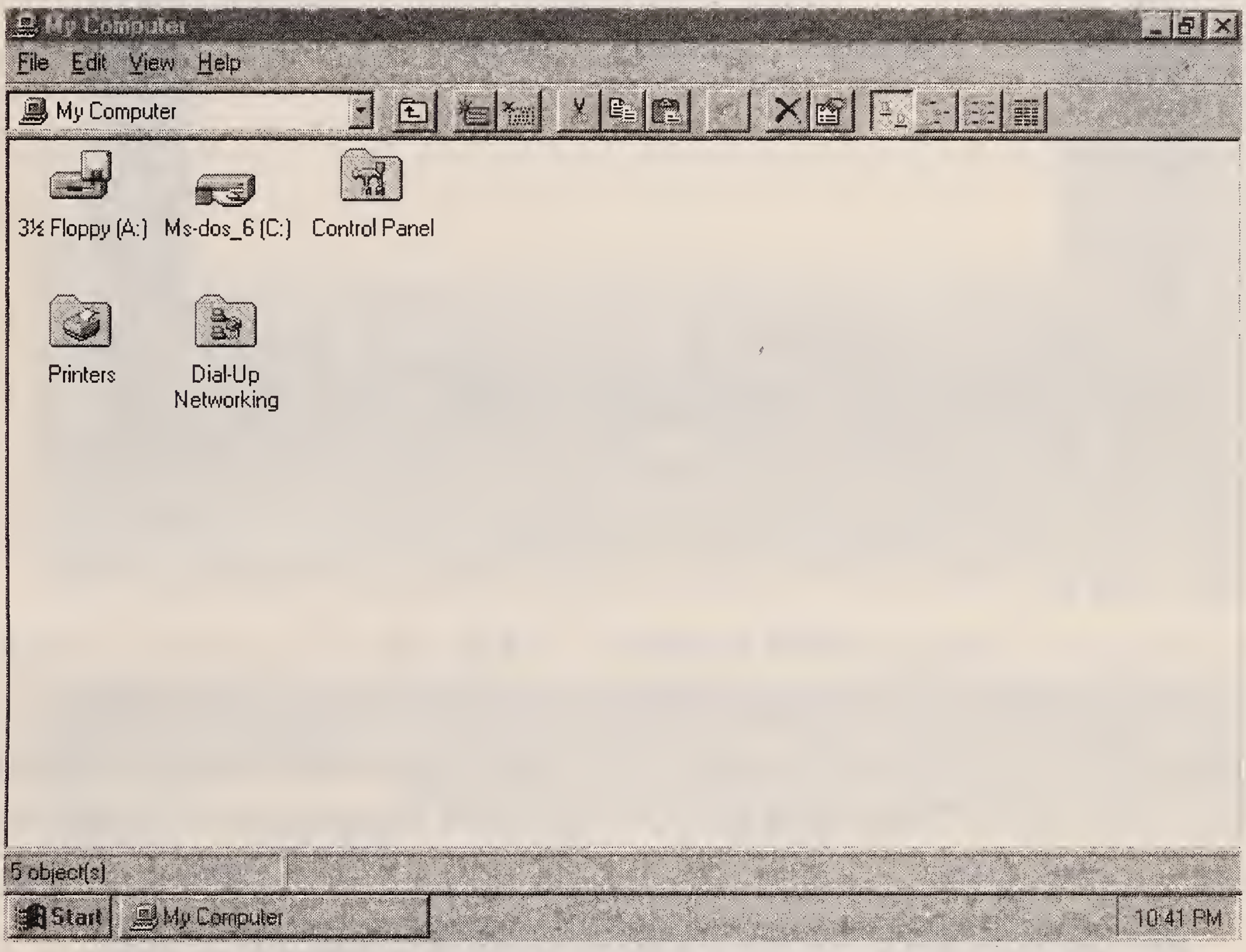

My Computer
File Edit View Help
My Computer
3½ Floppy (A:)
Ms-dos_6 (C:)
Control Panel
Printers
Dial-Up Networking
5 object(s)
Start
My Computer
10:41 PM

图 1.10 最大化窗口

（2）最小化窗口

单击窗口右上端左边的第一个小图标，即可最小化窗口，屏幕画面变成图 1.11。这时任务栏中“My Computer”按钮变成灰色，单击此处就可把窗口恢复到原来的大小。

最大化和最小化窗口也可以通过窗口左上端的控制按钮进行。单击控制按钮，再单击 Minimize 最小化窗口或 Maxmize 最大化窗口。

1.3.3 窗口大小的改变、移动及安排

（1）窗口大小的改变

把鼠标指向窗口的边框，这时鼠标光标变成带有双箭头的小指针，按住鼠标左按钮拖动鼠标直到窗口大小合乎用户的要求为止，然后释放鼠标按钮。如果用户把鼠标放在两边框的交界处，则通过拖动鼠标可以同时改变两边框的位置。

（2）窗口的移动

窗口的移动非常简单，把鼠标指向窗口顶部的标题栏，按住鼠标左按钮拖动鼠标到用户感到满意的位置，然后释放鼠标按钮，就可以把窗口移到新的位置。

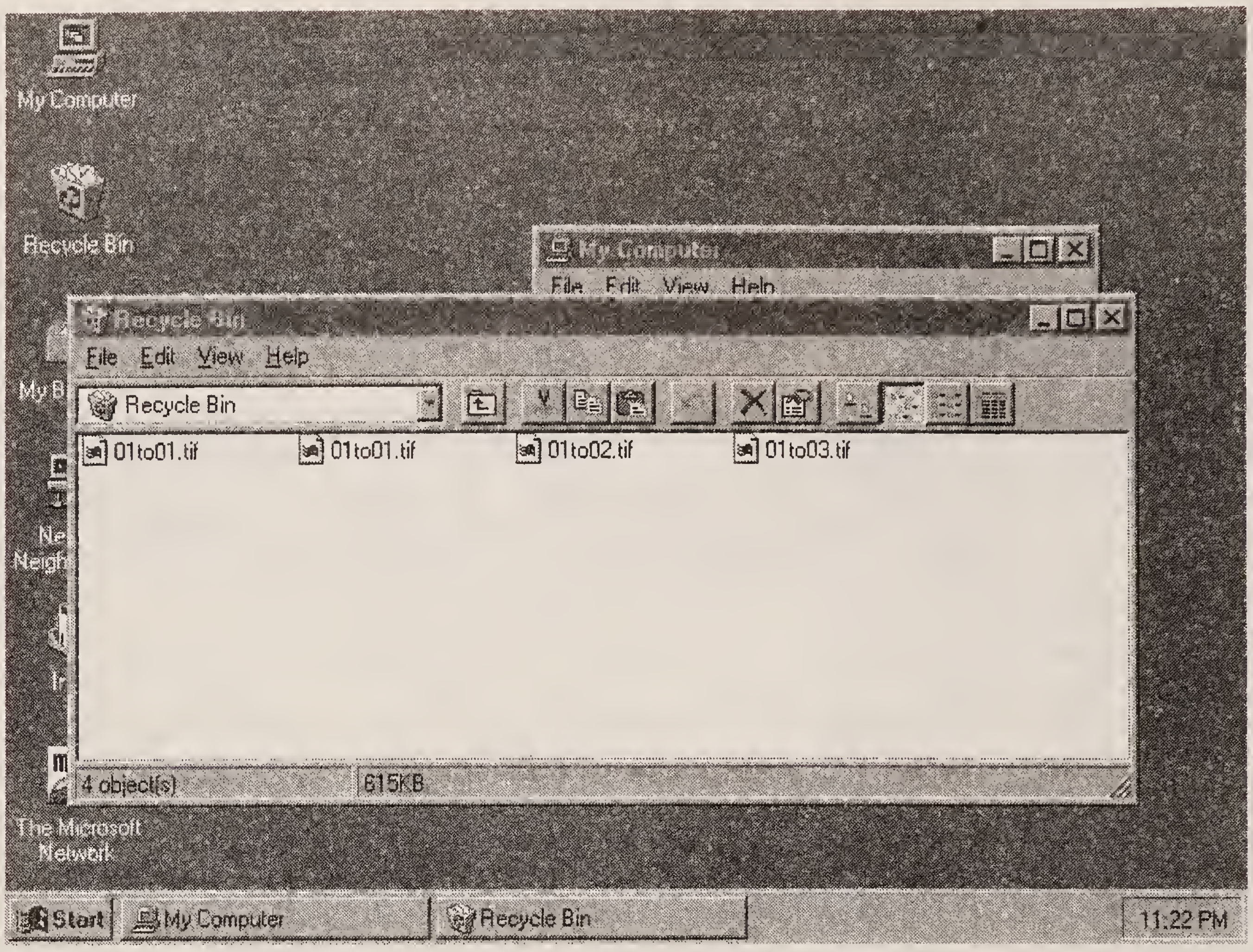

My Computer
Recycle Bin
The Microsoft Network
My Computer
File Edit View Help
Recycle Bin
File Edit View Help
Recycle Bin
01to01.tif
01to01.tif
01to02.tif
01to03.tif
4 object(s)
615KB
Start
My Computer
Recycle Bin
11:22 PM

图 1.11　最小化窗口

(3) 窗口的排列

鼠标右按钮可以方便地排列屏幕上的窗口。如双击桌面上“My Computer”和“Recycle Bin”两个图标，这时屏幕上出现重叠的两个窗口。用鼠标右按钮单击任务栏的空白处，这时出现一个菜单，见图 1.12。菜单上有许多命令，用户单击它们可以用不同的方式排列屏幕上出现的窗口。单击 Cascade 命令以重叠方式排列窗口，见图 1.13。单击 Tile Horizontally 命令以横向平铺方式排列，见图 1.14。单击 Tile Vertically 以纵向平铺方式排列窗口，见图 1.15。

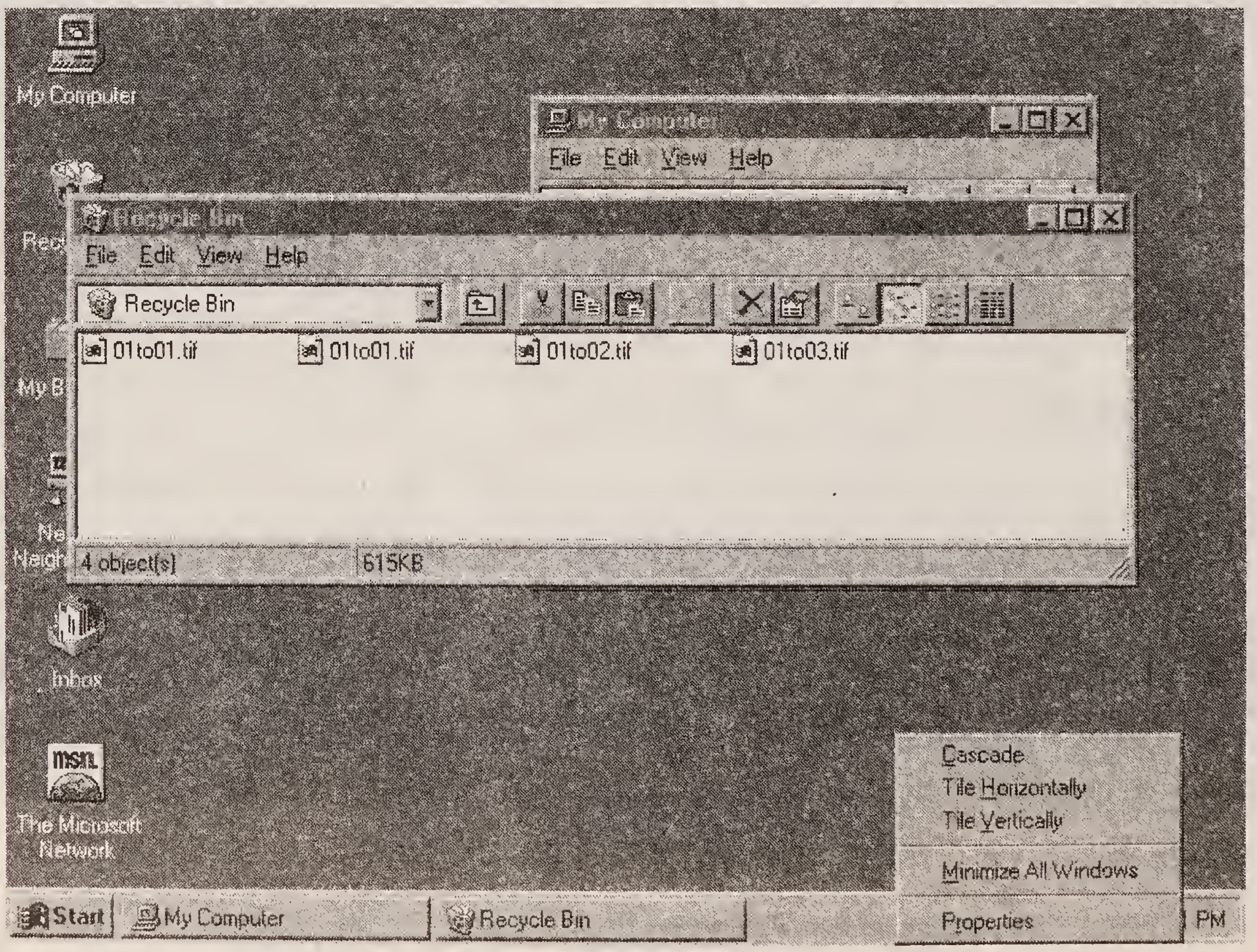

My Computer
My Computer
File Edit View Help
Recycle Bin
File Edit View Help
Recycle Bin
01to01.tif
01to01.tif
01to02.tif
01to03.tif
4 object(s)
615KB
Inbox
The Microsoft Network
Cascade
Tile Horizontally
Tile Vertically
Minimize All Windows
Properties
Start
My Computer
Recycle Bin
PM

图 1.12　选择菜单

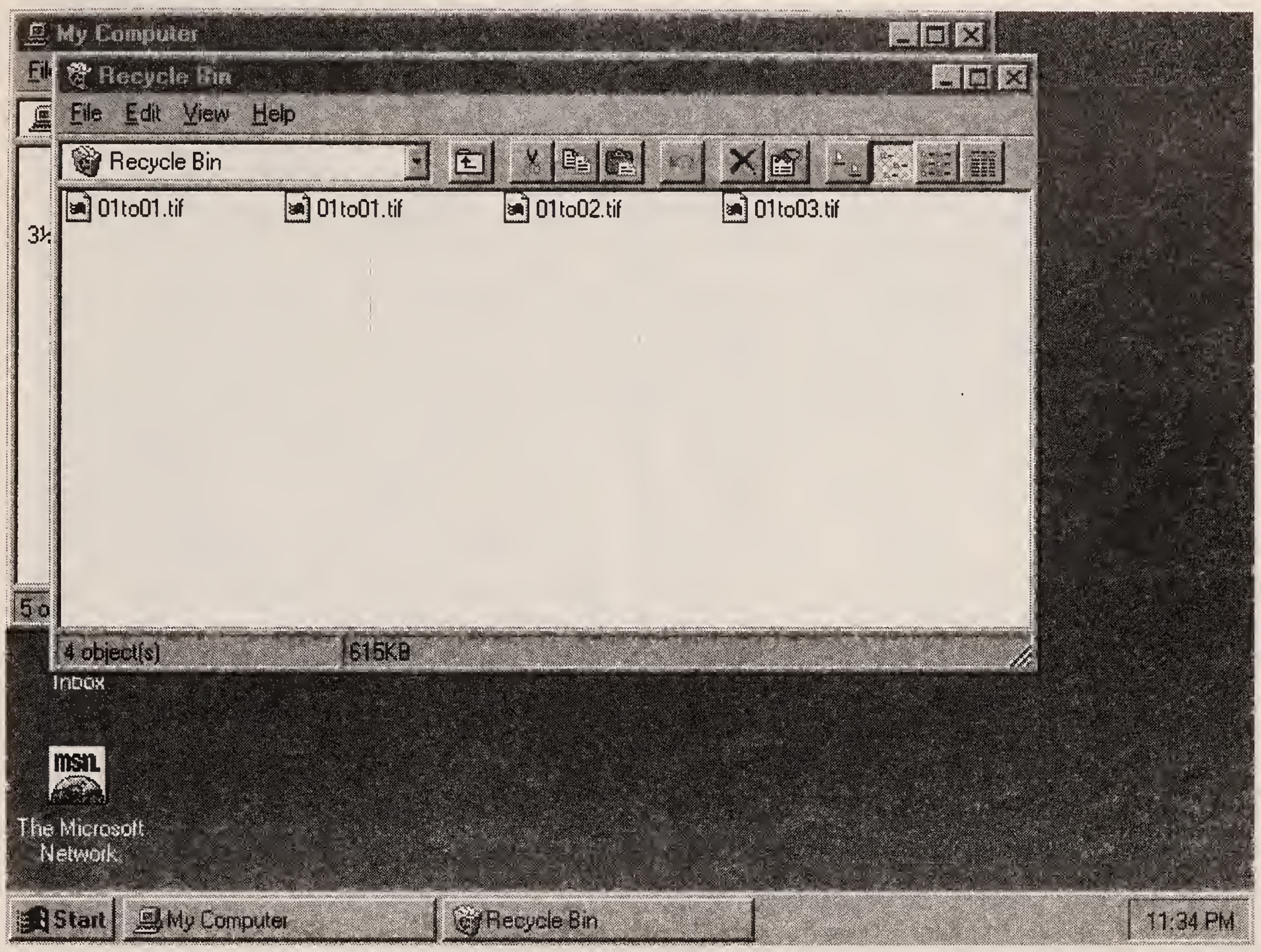

My Computer
Recycle Bin
File Edit View Help
Recycle Bin
01to01.tif
01to01.tif
01to02.tif
01to03.tif
4 object(s)
615KB
Inbox
The Microsoft Network
Start
My Computer
Recycle Bin
11:34 PM

图 1. 13 重叠方式排列窗口

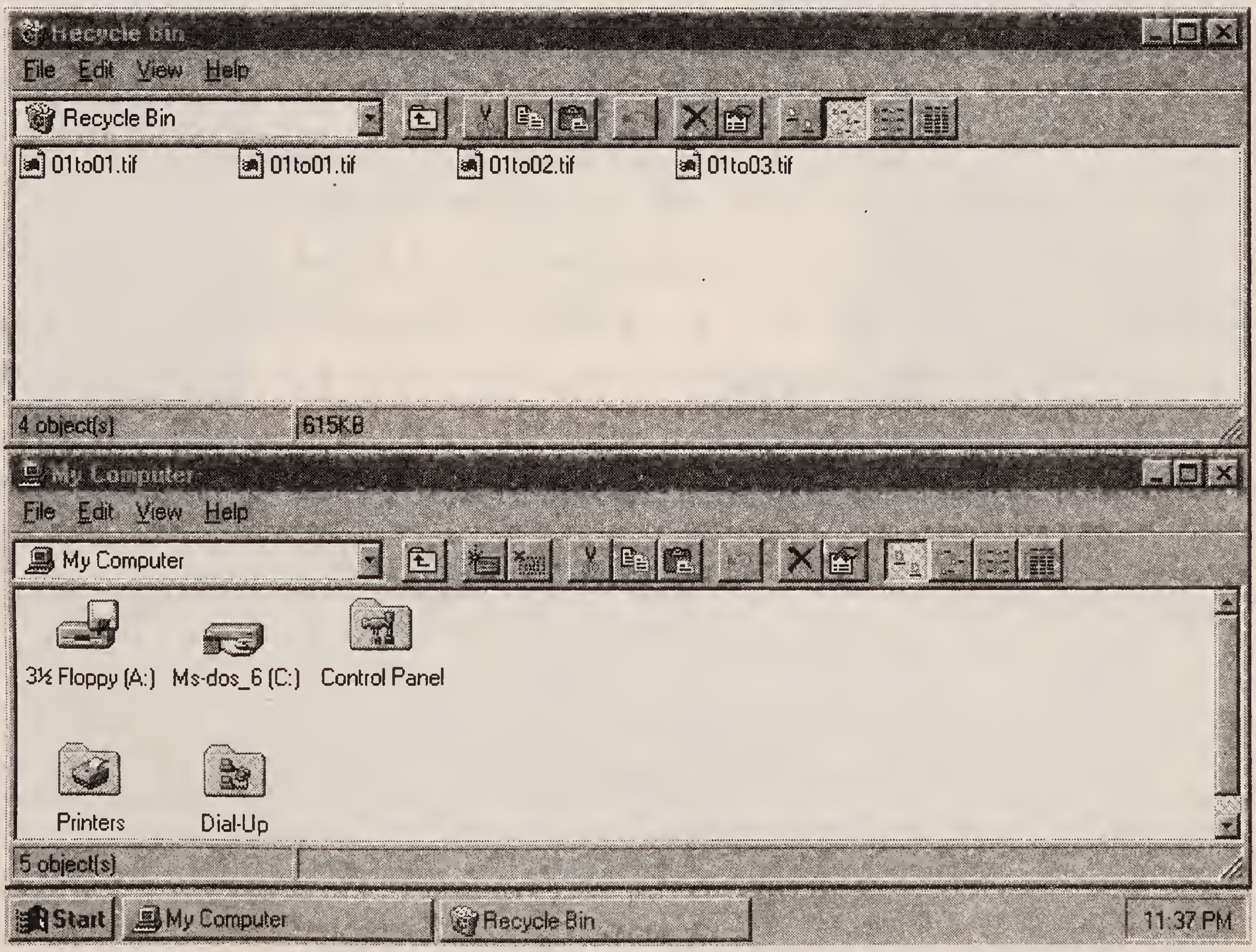

Recycle Bin
File Edit View Help
Recycle Bin
01to01.tif
01to01.tif
01to02.tif
01to03.tif
4 object(s)
615KB
My Computer
File Edit View Help
My Computer
3½ Floppy (A:)
Ms-dos_6 (C:)
Control Panel
Printers
Dial-Up
5 object(s)
Start
My Computer
Recycle Bin
11:37 PM

图 1. 14 横向平铺排列窗口

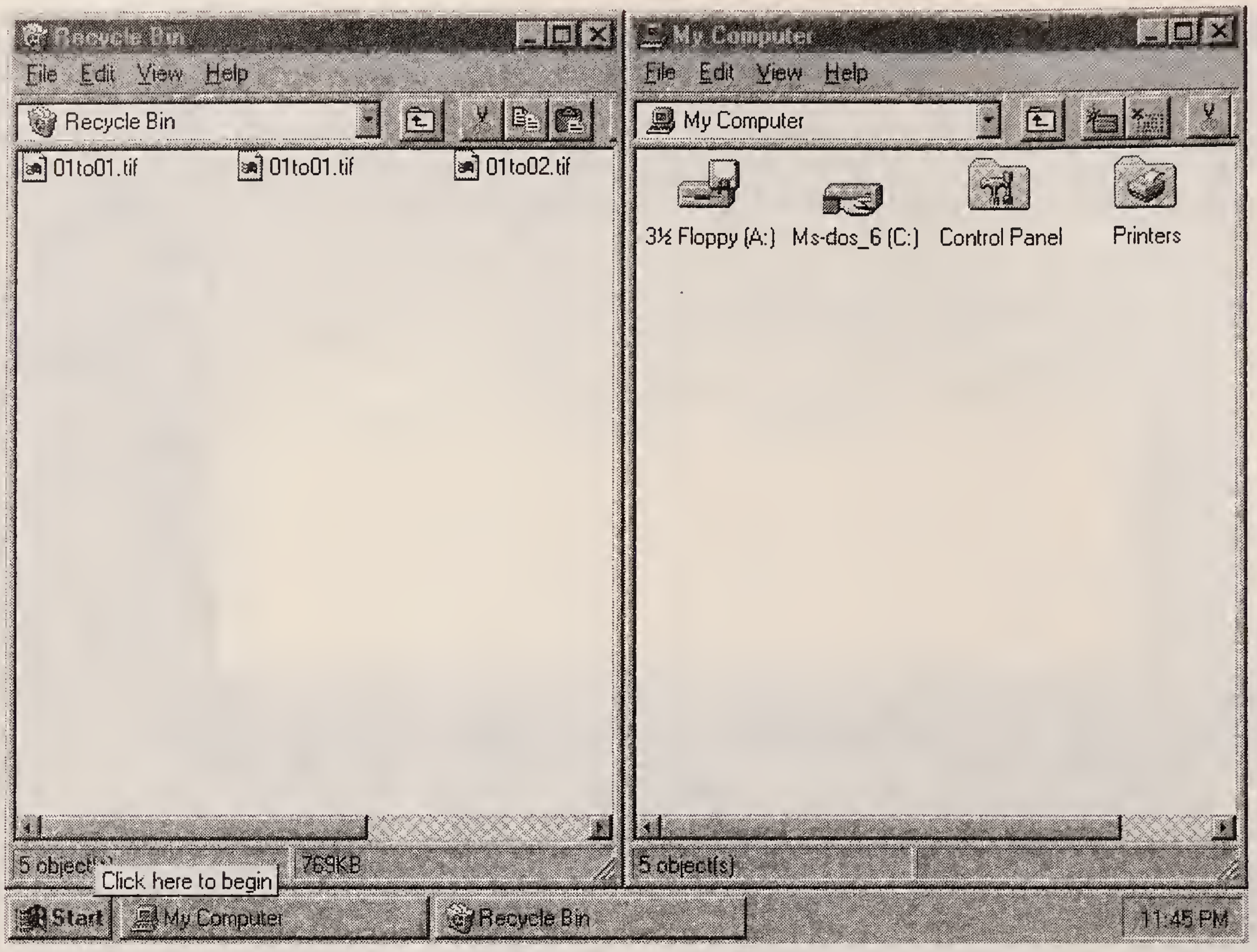

Recycle Bin
File Edit View Help
Recycle Bin
01to01.tif
01to01.tif
01to02.tif
5 object(s)
Click here to begin
769KB
My Computer
File Edit View Help
My Computer
3½ Floppy (A:)
Ms-dos_6 (C:)
Control Panel
Printers
5 object(s)
Start
My Computer
Recycle Bin
11:45 PM

图 1.15 纵向平铺排列窗口

1.3.4 图标的排列和显示方式的改变

在启动 Windows 95 时，桌面上的图标被自动地放置在桌面的左边缘。就像用户能够移动桌面上的窗口一样，用户也能够移动桌面上的图标到桌面的任意位置上。操作方法同移动窗口相似，也是拖动图标到用户要放置的位置上，然后释放鼠标按钮。

注意　也许用户按照上述方法不能够移动图标，这可能是因为用户的环境中设置了“Auto Arrage”(自动排列)一项。这时用户只需在桌面上单击鼠标右按钮，此时显示出一个菜单，用户把鼠标指向 Arrange Icons，然后单击 Auto Arrange 命令，见图 1.16。

用户也能够用鼠标右按钮改变显示方式。例如，用户双击“My Computer”图标，打开“My Computer”窗口，用鼠标右按钮单击窗口的空白处，这时出现一菜单，用鼠标指向 View 选项打开它的级联菜单(见图 1.17)。单击不同的命令就可以用不同的方式显示“My Computer”窗口中图标：

“Large Icons”	以大图标方式显示
“Small Icons”	以小图标方式显示
“List”	以列表方式显示
“Details”	以详细资料方式显示

提示　在 Windows 95 中，在某一对象上单击鼠标右按钮，大多数都会弹出一个菜单，称为快捷菜单，上面的选项用来完成该对象的操作。如在磁盘对象上单击右按钮，显示一个有关磁盘复制、格式化、共享和属性设置等操作的快捷菜单。关于鼠标右按钮的其他功能可以参考本社出版的《Windows 95 从入门到精通》一书的有关内容。

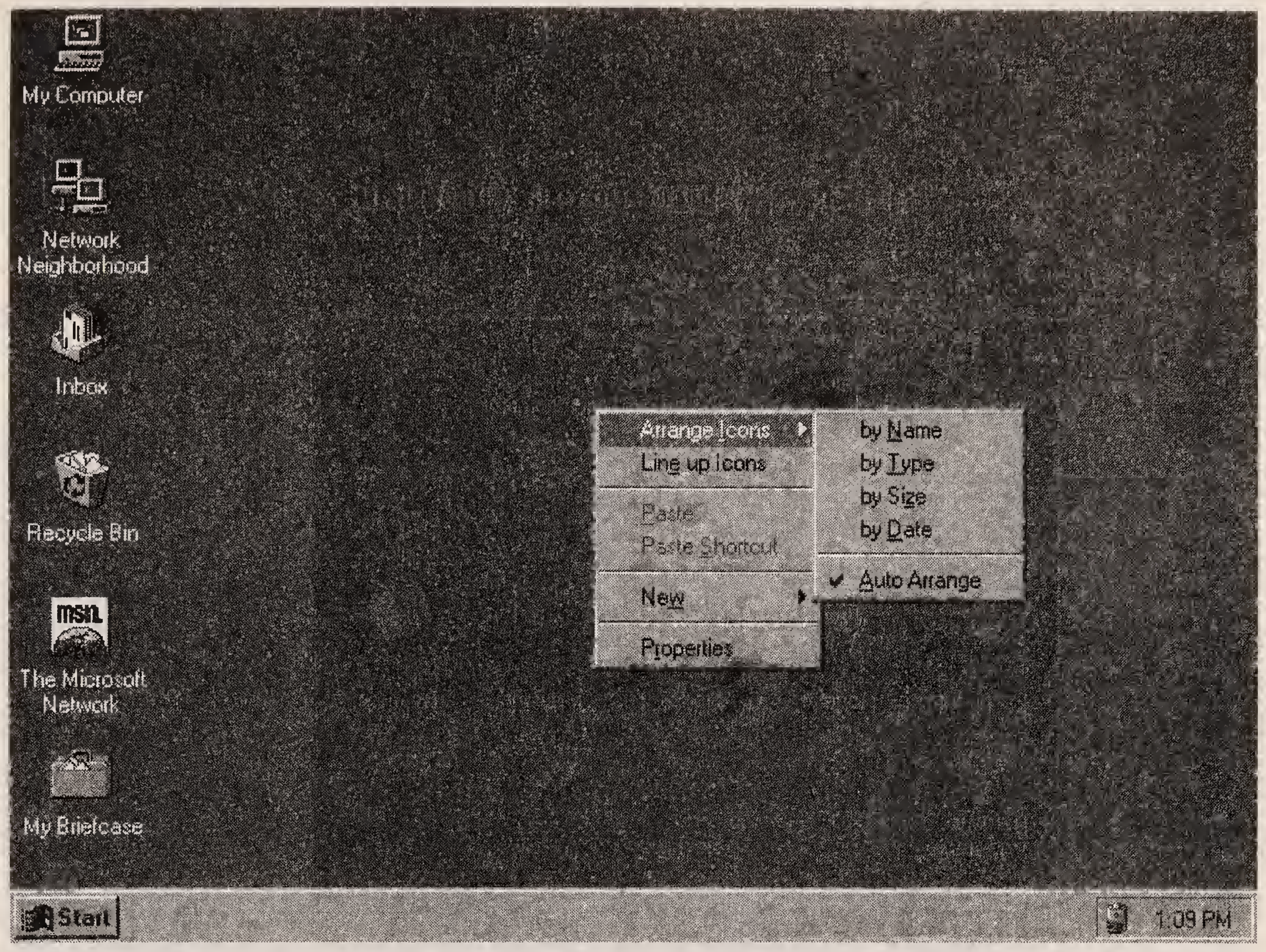
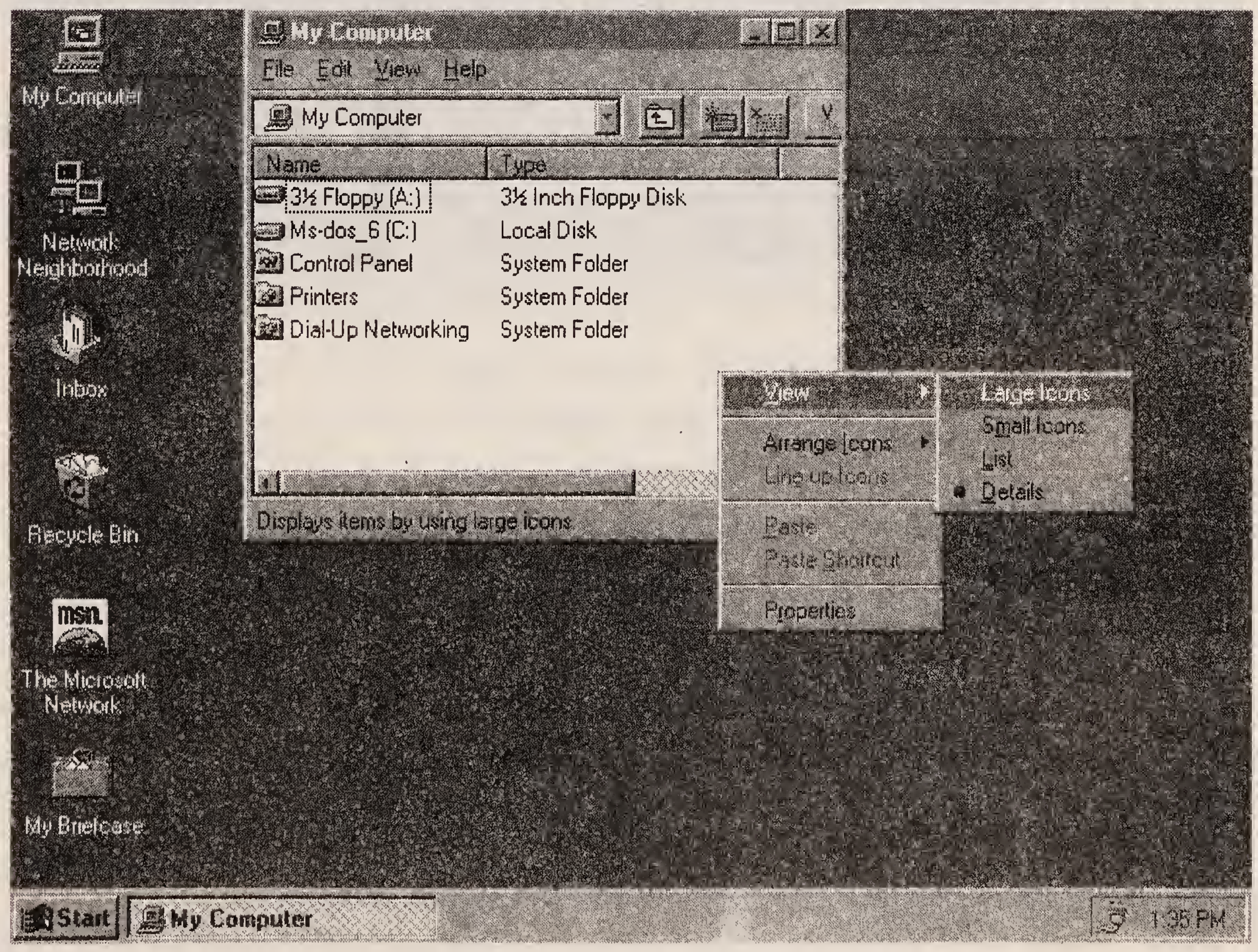

My Computer
Network Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Arrange Icons
Line up Icons
Paste
Paste Shortcut
New
Properties
by Name
by Type
by Size
by Date
✓ Auto Arrange
Start
1:09 PM

图 1.16 消除自动排列(Auto Arrange)


My Computer
File Edit View Help
My Computer
Network Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
View
Arrange Icons
Line up Icons
Paste
Paste Shortcut
Properties
Large Icons
Small Icons
List
● Details
Start
My Computer
1:35 PM

Name	Type
3½ Floppy (A:)	3½ Inch Floppy Disk
Ms-dos_6 (C:)	Local Disk
Control Panel	System Folder
Printers	System Folder
Dial-Up Networking	System Folder

Displays items by using large icons

图 1.17 显示方式菜单

1.4 程序的运行和安装

1.4.1 Start 菜单的使用

Start 运行 Windows 95 时，在屏幕左下角有一 Start 按钮，单击此按钮，用户可以看见一菜单，见图 1.18，它包括以下选项：

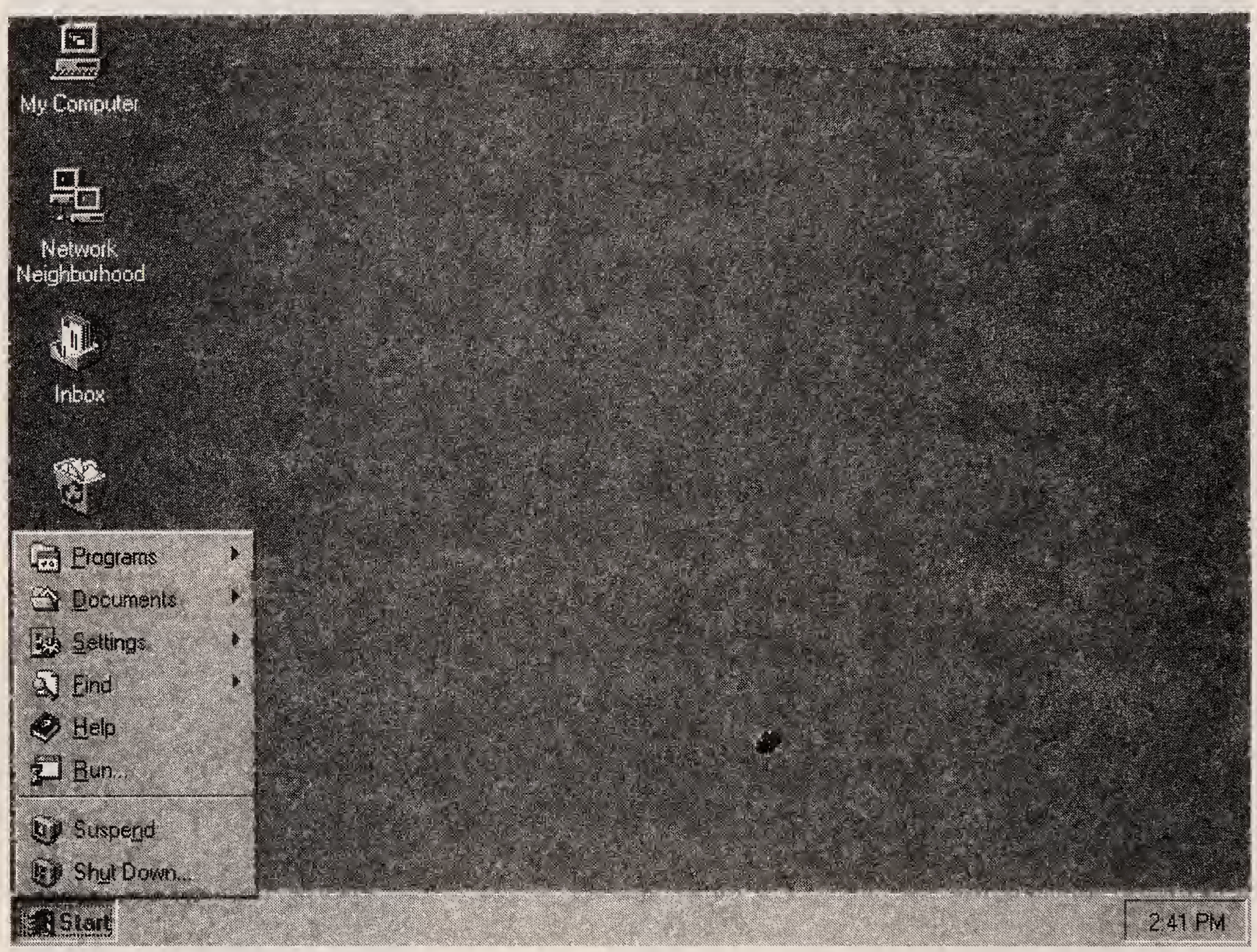

My Computer
Network Neighborhood
Inbox
Programs
Documents
Settings
Find
Help
Run...
Suspend
Shut Down...
Start
2:41 PM

图 1.18 打开 Start 菜单

Programs：	显示可运行程序或组的清单。
Documents：	显示最近被打开的文档名，用户可以单击文档名打开文档。
Settings：	用来改变或者查看 Windows 95 的设置。
Find：	用来发现用户电脑或网络中有关的文件或文件夹等资源。
Help：	提供联机帮助信息。
Run：	用来运行程序或打开文件夹。
Shut Down：	用来在关机之前关闭 Windows 95、重新启动计算机、退出网络重新进入 Windows 95 或者从 Windows 95 返回到 DOS 状态下。

另外，根据计算机的设置不同，菜单上还可能出现其他一些选项。

提示 Start 菜单可能与用户在 Windows 早期版本上所见的菜单有点不同，有的选项的右边有一个小箭头符号，像“Programs”、“Settings”等，这说明它们有级联菜单。用户不必单击它们而显示级联菜单(尽管用户能够)，只需用鼠标指向箭头，然后平移鼠标，这时级联菜单就会显示出来，见图 1.19。

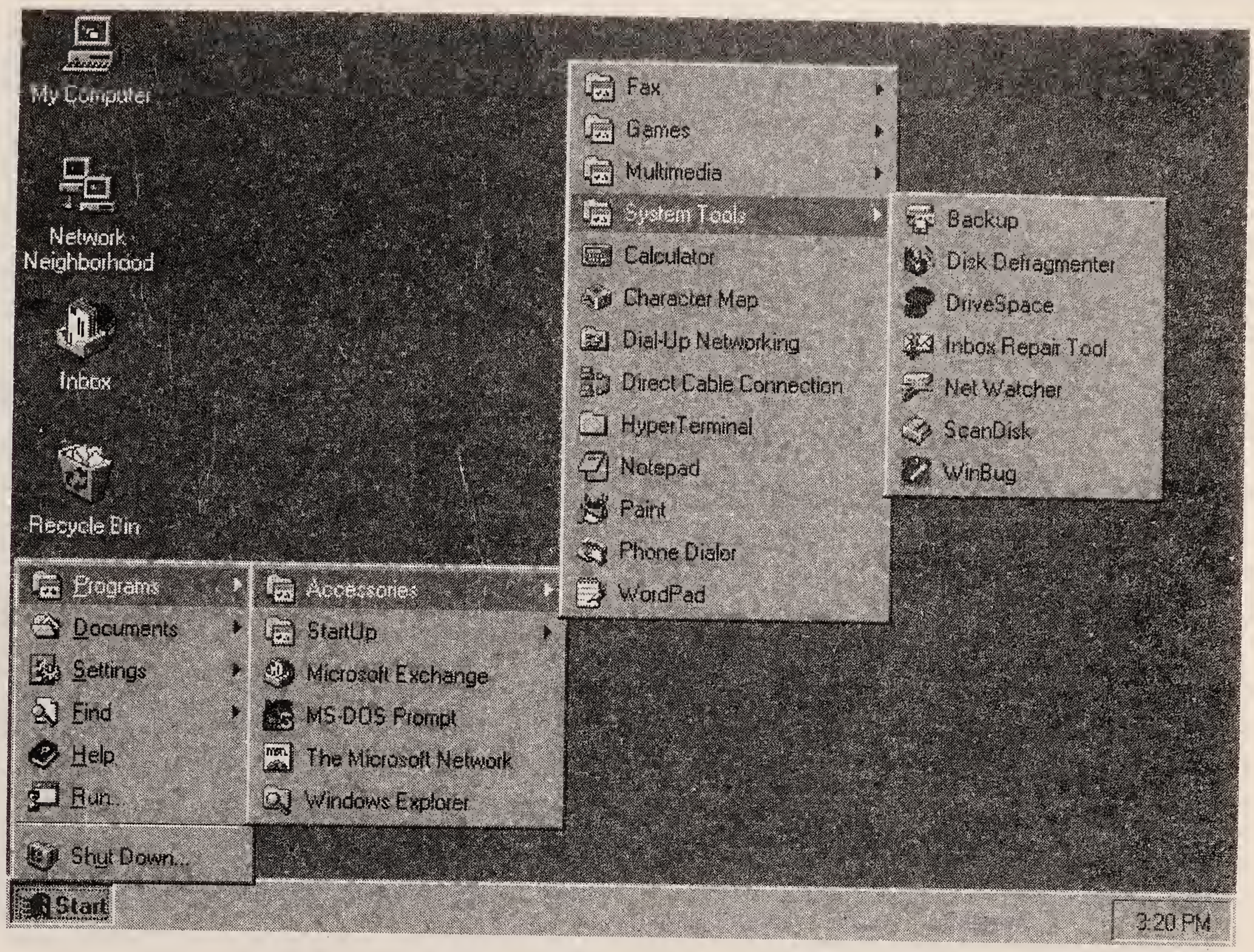

My Computer
Network Neighborhood
Inbox
Recycle Bin
Programs
Documents
Settings
Find
Help
Run...
Shut Down...
Start
Accessories
StartUp
Microsoft Exchange
MS-DOS Prompt
The Microsoft Network
Windows Explorer
Fax
Games
Multimedia
System Tools
Calculator
Character Map
Dial-Up Networking
Direct Cable Connection
HyperTerminal
Notepad
Paint
Phone Dialer
WordPad
Backup
Disk Defragmenter
DriveSpace
Inbox Repair Tool
Net Watcher
ScanDisk
WinBug
3:20 PM

图 1.19 怎样打开级联菜单

1.4.2 程序的运行

在 Windows 95 中，用户运行一个应用程序的方法有多种。用户可以根据需要而选择。运行一个应用程序最一般的方法利用 Start 菜单。为了帮助跟踪用户所有程序，Windows 95 把程序存放在文件夹中，文件夹就像 Windows 95 前期版本中的目录的作用，主要是为了方便用户管理。

用户打开 Start 菜单，把鼠标指向 Programs，按照前述方法打开级联菜单直到用户想运行的应用程序出现，单击应用程序图标，就可以运行该应用程序，见图 1.20。

例如，用户单击 Calculator 图标就可以运行计算器程序，见图 1.21。

用户也可以单击 Start 菜单的 Run 命令来运行程序，这时出现 Run 对话框，在对话框中键入用户需要运行程序的完整路径及文件名，然后单击 OK 按钮，见图 1.22。

如果用户忘记了文件名，可以用对话框底部 Browse 命令按钮来查找。单击 Browse 按钮，屏幕上出现查找窗口，见图 1.23。

在查找窗口中打开“Lookin”下拉式列表框（单击框中右侧的箭头），单击用户要运行程序所在的驱动器或网上邻居，见图 1.24。

双击包含要运行程序存在的文件夹，打开文件夹，最后双击要运行的程序，见图 1.25。

这时屏幕重新返回到 Run 对话框，并且把用户所选的程序的路径和文件名插入到 Run 对话框中，单击 OK 按钮就可运行所选择的程序。见图 1.26。

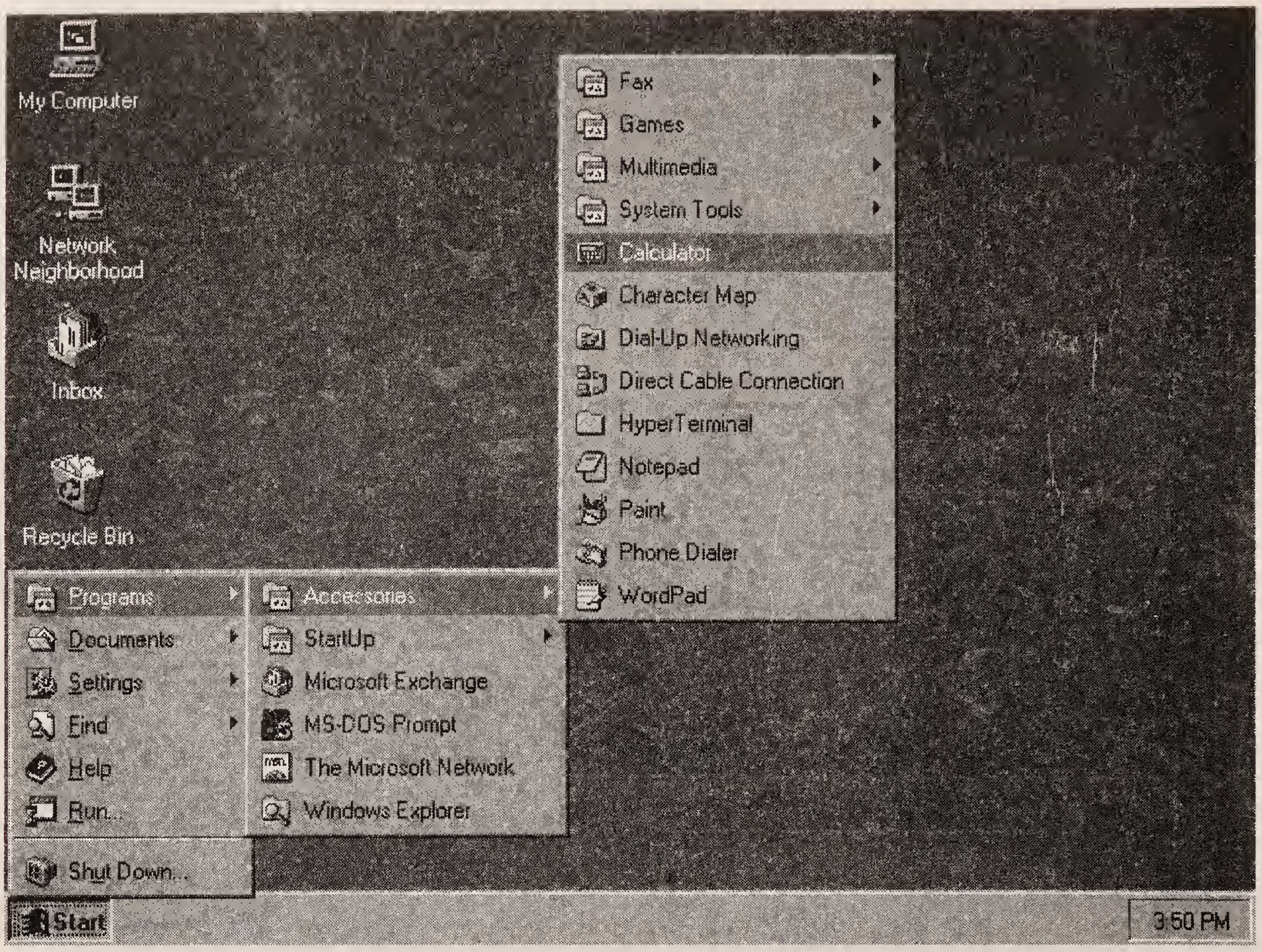

My Computer
Network Neighborhood
Inbox
Recycle Bin
Fax
Games
Multimedia
System Tools
Calculator
Character Map
Dial-Up Networking
Direct Cable Connection
HyperTerminal
Notepad
Paint
Phone Dialer
WordPad
Programs
Documents
Settings
Find
Help
Run...
Shut Down...
Accessories
StartUp
Microsoft Exchange
MS-DOS Prompt
The Microsoft Network
Windows Explorer
Start
3:50 PM

图 1.20　选择要运行的程序

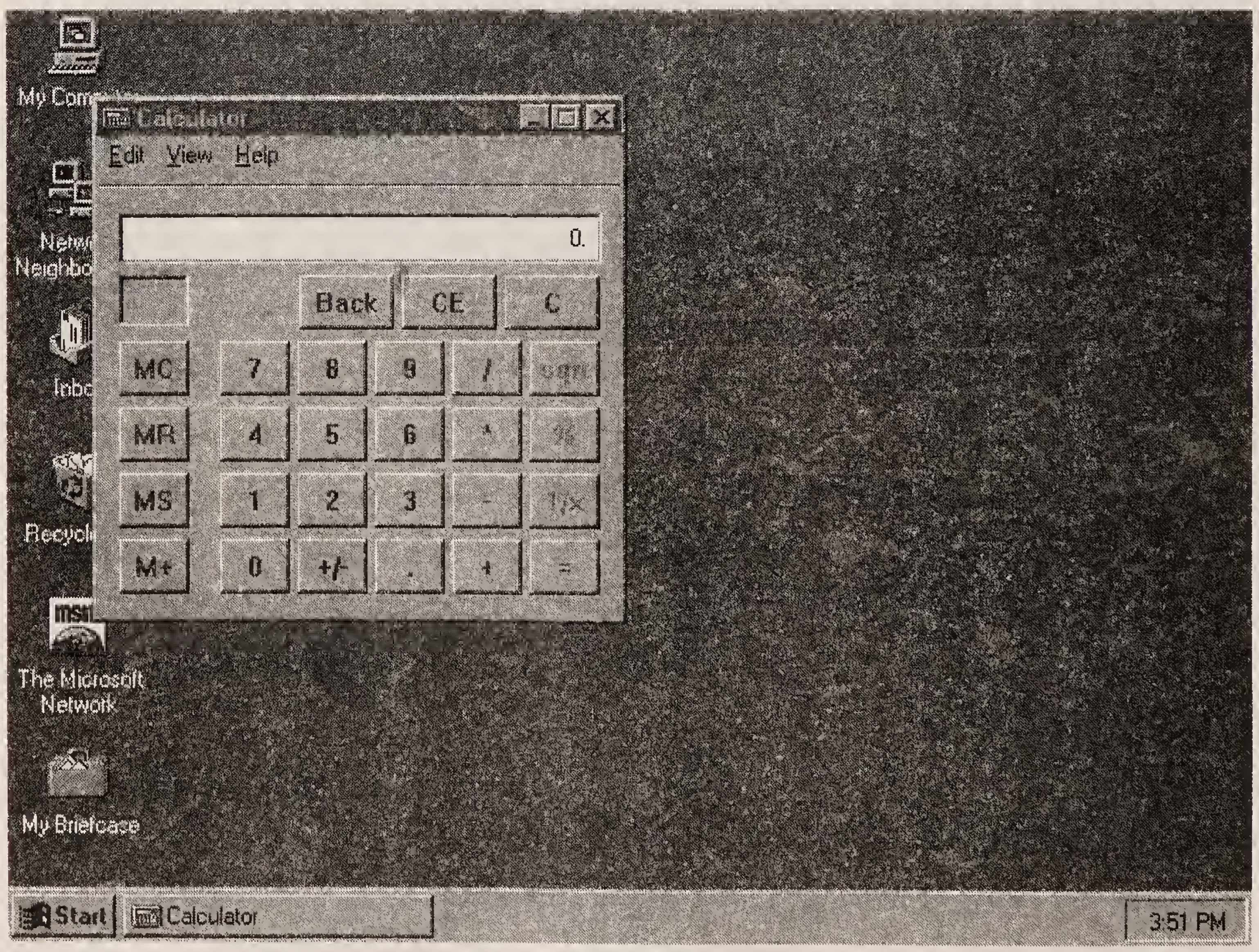
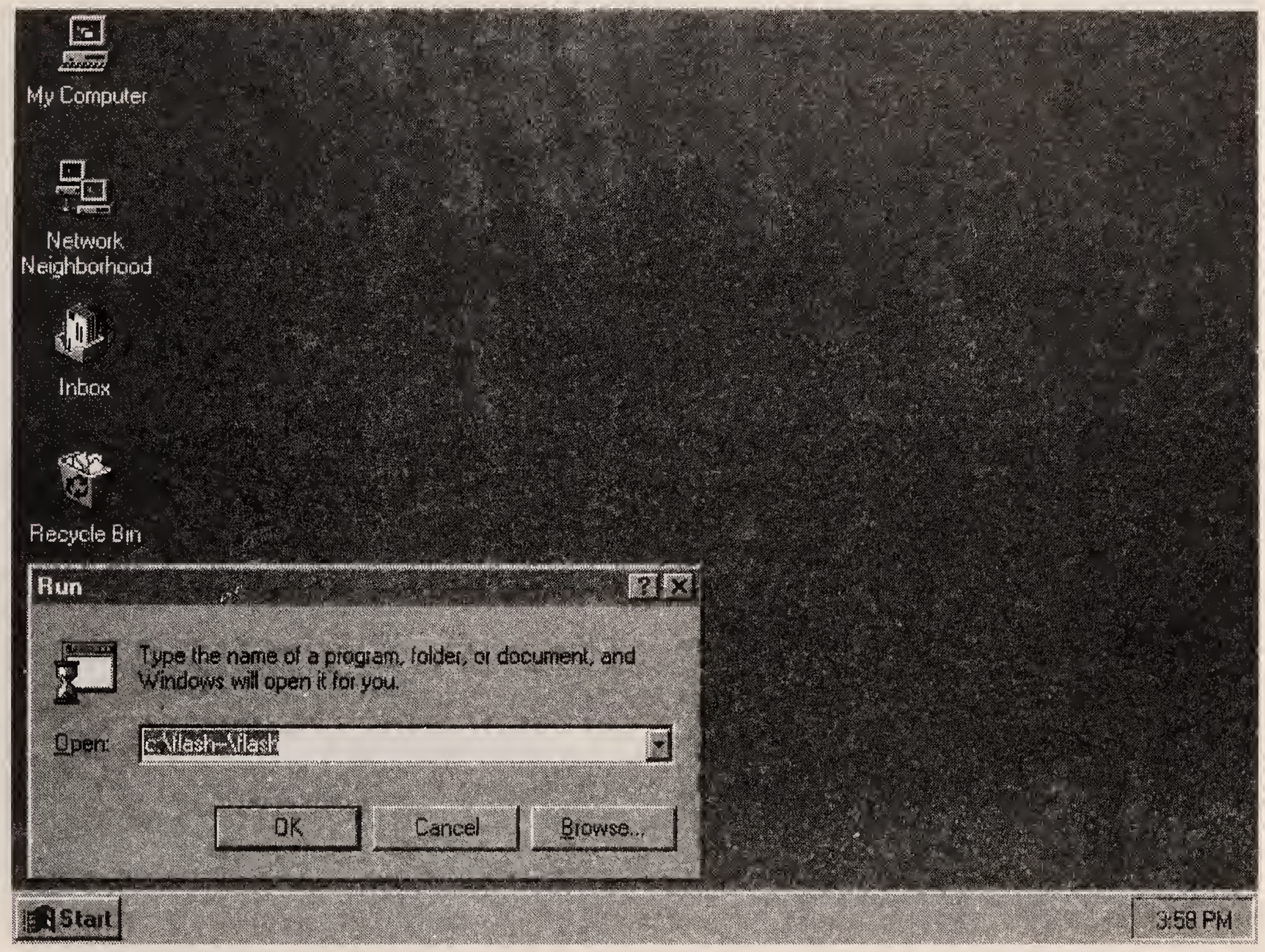

Calculator
Edit View Help
0.
Back CE C
MC 7 8 9 / sqrt
MR 4 5 6 * %
MS 1 2 3 - 1/x
M+ 0 +/- . + =
The Microsoft Network
My Briefcase
Start
Calculator
3:51 PM

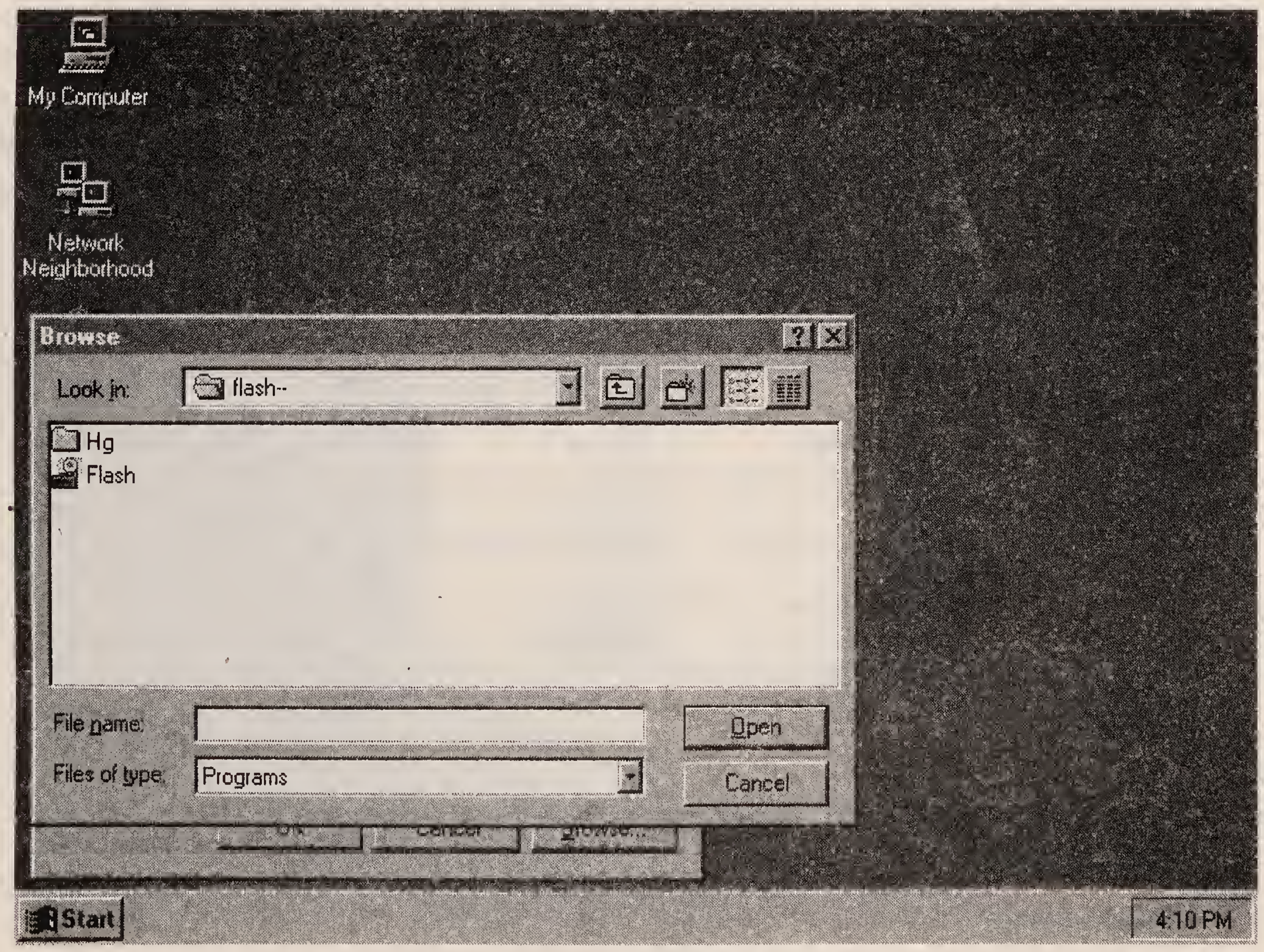
图 1.21　运行计算器程序


My Computer
Network Neighborhood
Inbox
Recycle Bin
Run
OK
Cancel
Browse...
Start
3:58 PM

Type the name of a program, folder, or document, and Windows will open it for you.

Open: c:\flash--\flash

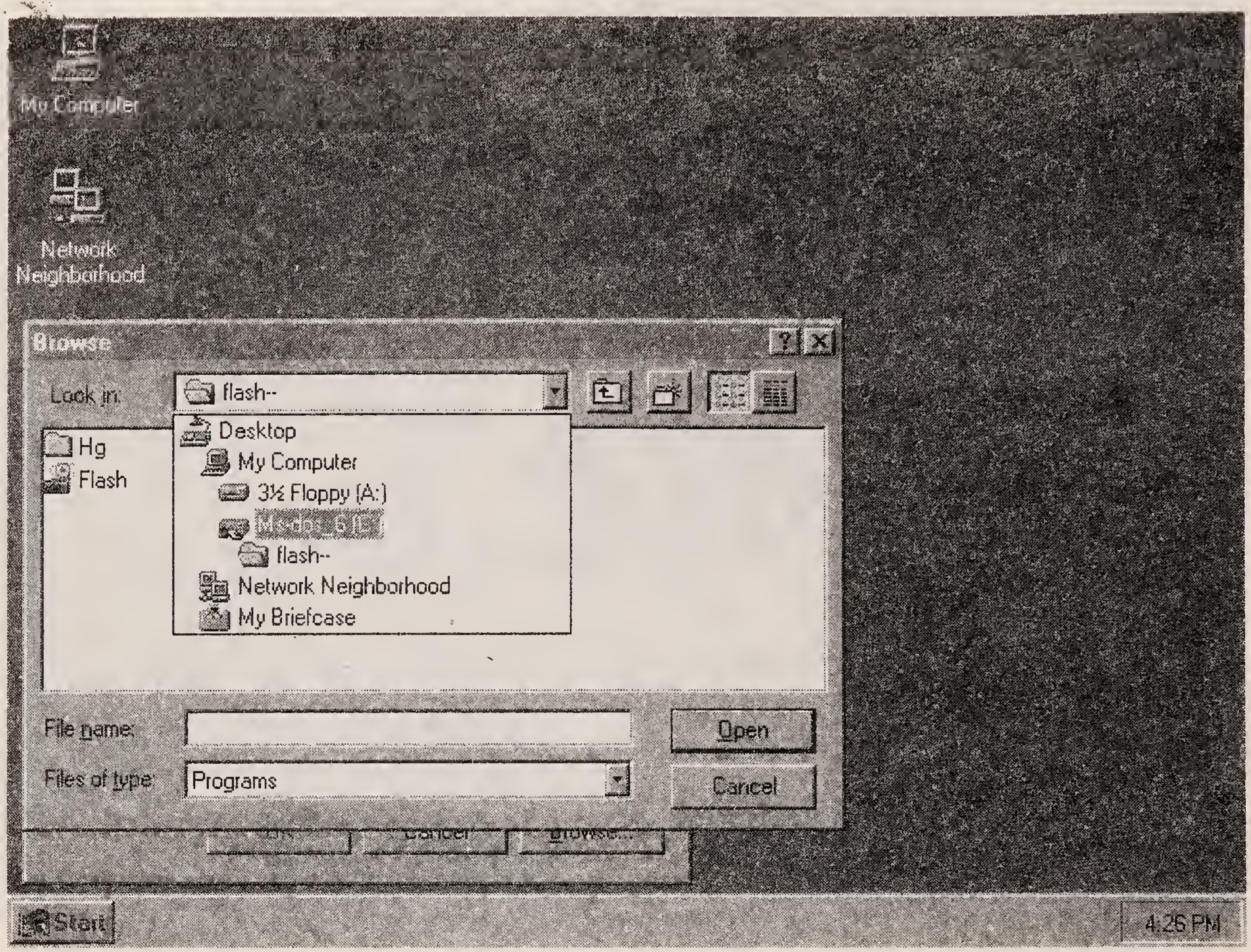
图 1. 22　Run 对话框


My Computer
Network Neighborhood
Browse
Hg
Flash
File name:
Open
Cancel
Start
4:10 PM

Look in: flash--

Files of type: Programs

图 1. 23　查找窗口

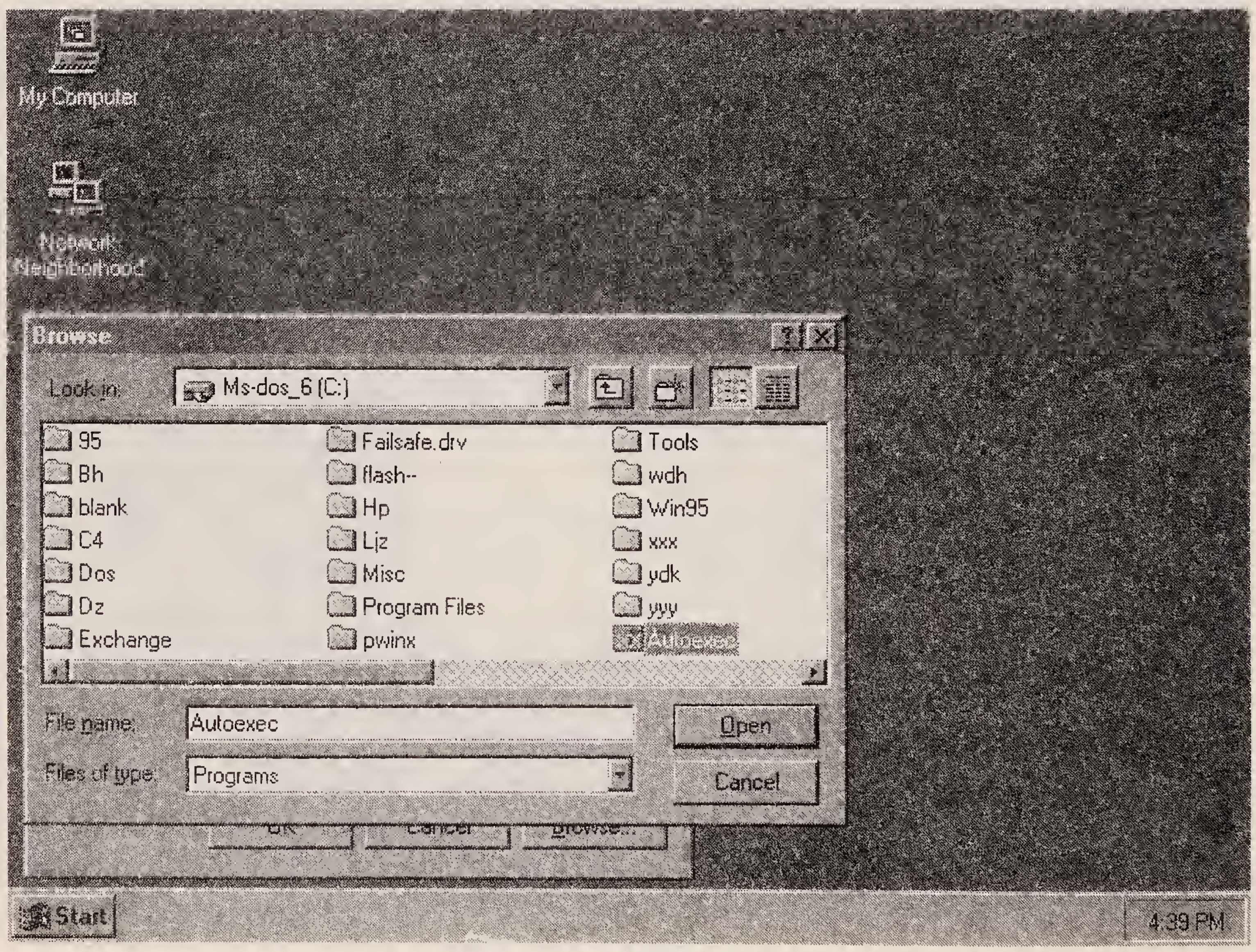


Look in: flash--

Files of type: Programs

My Computer
Network Neighborhood
Browse
Hg
Flash
Desktop
My Computer
3½ Floppy (A:)
Ms-dos_6 (C:)
flash--
Network Neighborhood
My Briefcase
File name:
Open
Cancel
Start
4:26 PM

图 1.24　选择驱动器



Look in: Ms-dos_6 (C:)

File name: Autoexec

Files of type: Programs

My Computer
Network Neighborhood
Browse
95
Bh
blank
C4
Dos
Dz
Exchange
Failsafe.drv
flash--
Hp
Ljz
Misc
Program Files
pwinx
Tools
wdh
Win95
xxx
ydk
yyy
Autoexec
Open
Cancel
Start
4:39 PM

图 1.25　选择查找程序

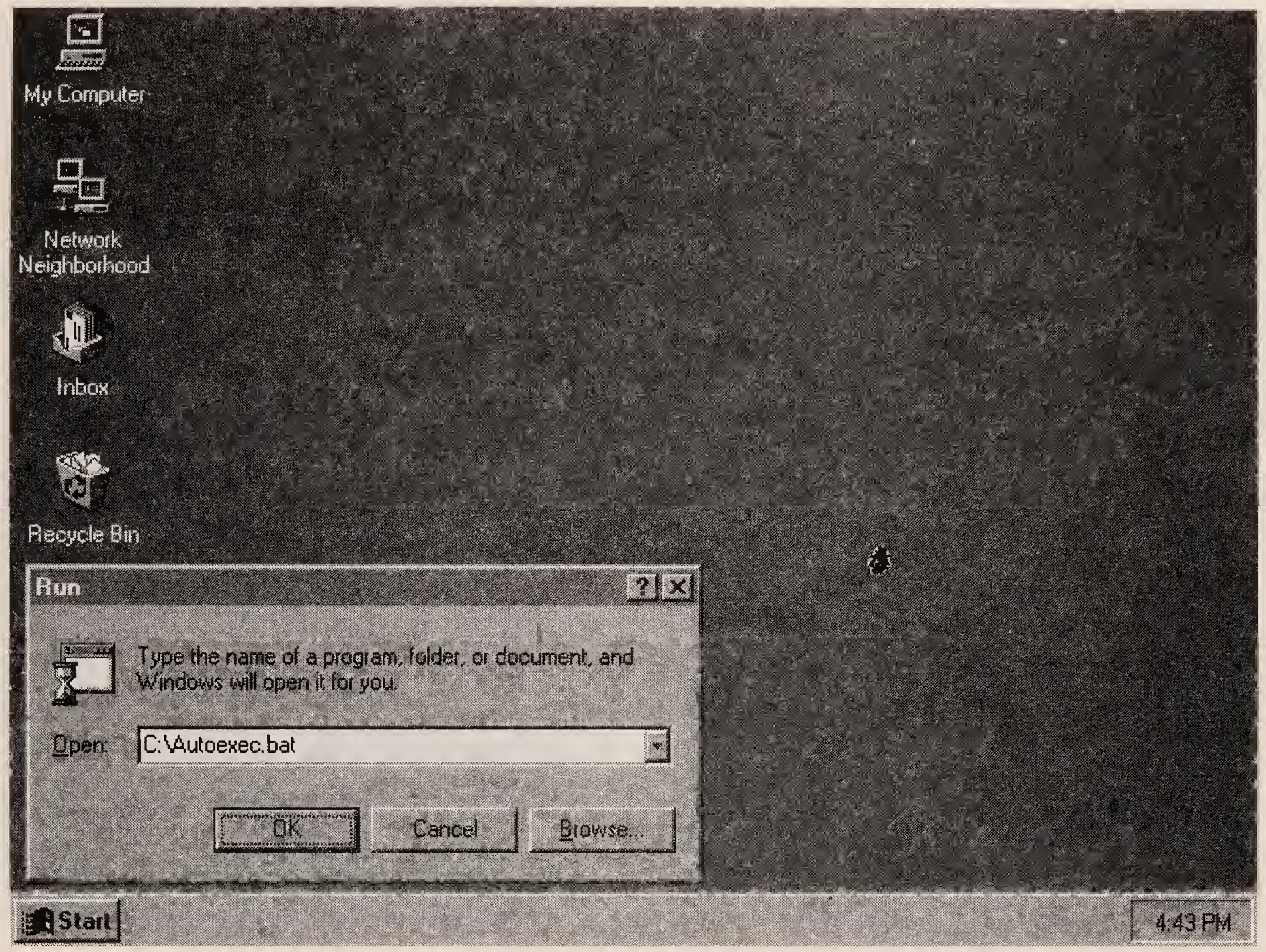

My Computer
Network Neighborhood
Inbox
Recycle Bin
Run
Type the name of a program, folder, or document, and Windows will open it for you.
Open:
C:\Autoexec.bat
OK
Cancel
Browse...
Start
4:43 PM

图 1.26 在 Run 对话框中插入文件名及路径

1.4.3 程序的安装

安装一个应用程序有两种方法，一种是用 Start 菜单的 Run 命令，另一种是用 Start 菜单 Settings 命令。

(1) 用 Run... 命令安装程序

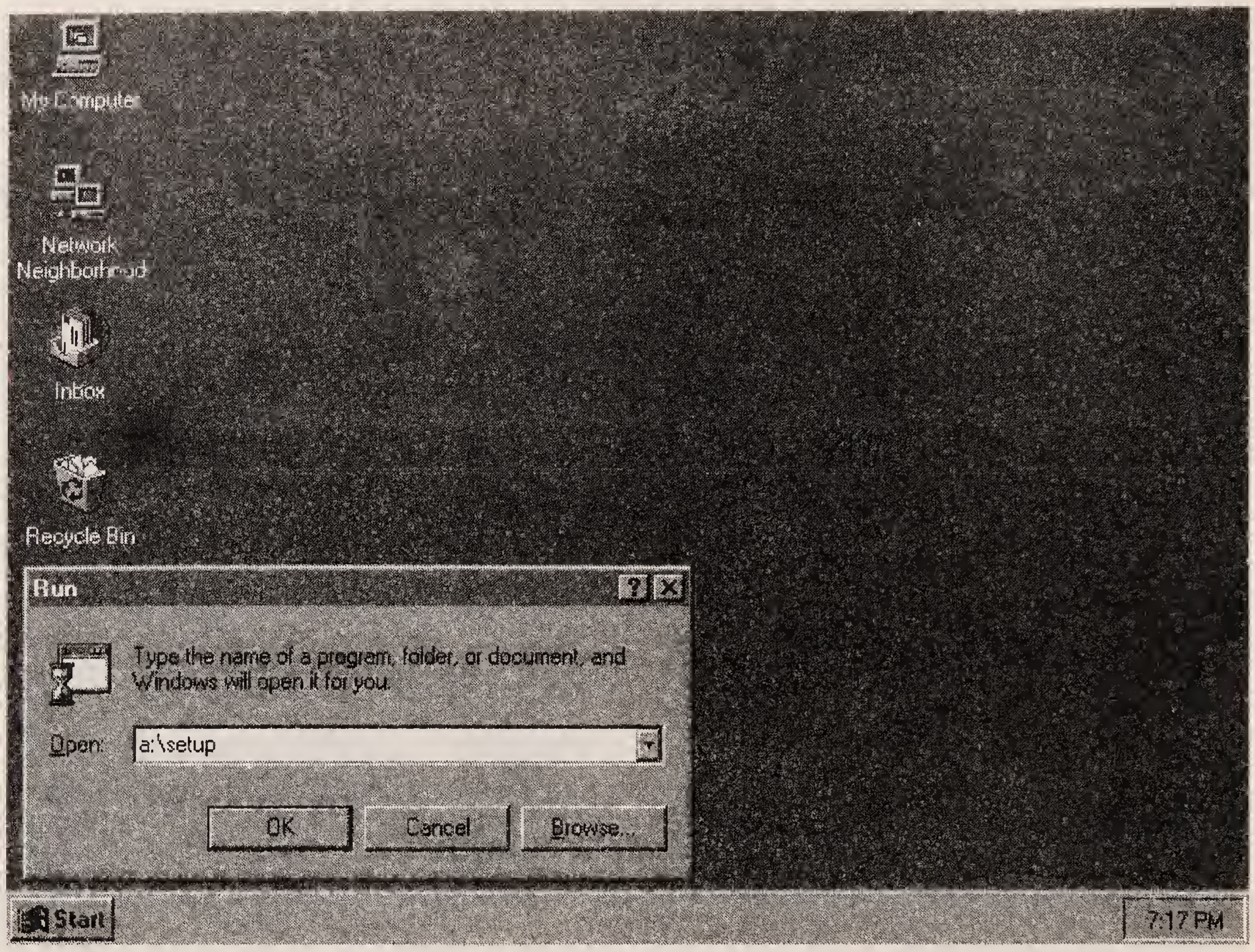
打开 Start 菜单，单击 Run 命令，在对话框中键入安装程序的路径和文件名。例如，A:\setup，再单击 OK 按钮，见图 1.27。然后根据屏幕提示的有关信息就可完成程序的安装。

(2) 用 Settings 选项安装程序

1) 打开 Start 菜单，把鼠标指向 Settings 打开它的级联菜单，然后单击 Control Panel 文件夹的图标，此时屏幕上出现控制面板窗口，见图 1.28。

2) 双击 Add/Remove Programs 项的图标，这时出现 Add/Remove Programs 对话框(见图 1.29)，单击 Install 命令按钮。这时屏幕上显示图 1.30，提示把含有安装程序的磁盘插入软驱或 CD-ROM 驱动器上。

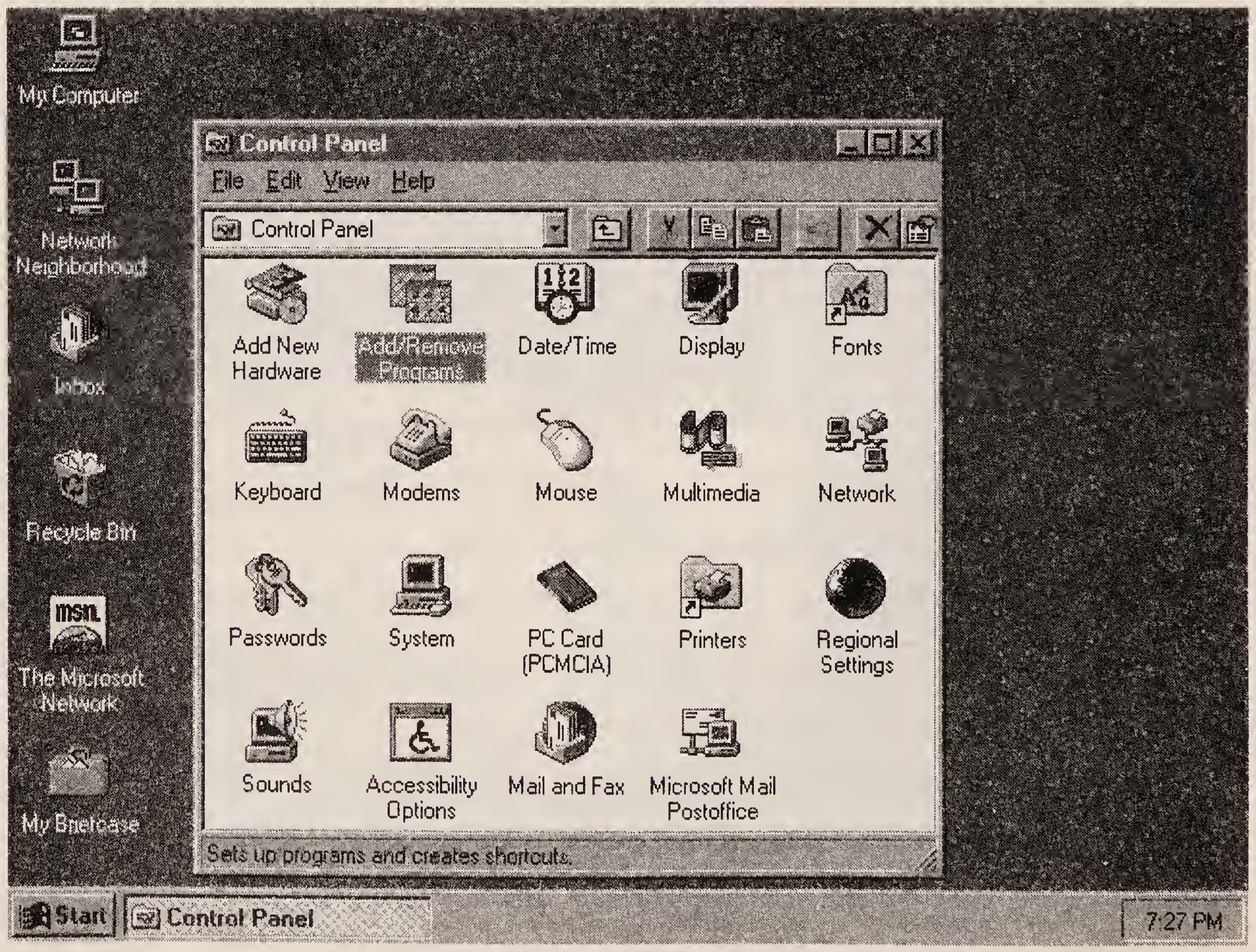
3) 在图 1.30 上单击 Next 命令按钮，出现一对话框，见图 1.31。键入安装程序的路径及文件名。例如，A:\setup.EXE。然后单击 Finish 命令按钮就执行了安装程序，根据屏幕提示的操作信息即可完成程序的安装工作。


My Computer
Network Neighborhood
Inbox
Recycle Bin
Run
OK
Cancel
Browse...
Start
7:17 PM

Type the name of a program, folder, or document, and Windows will open it for you.

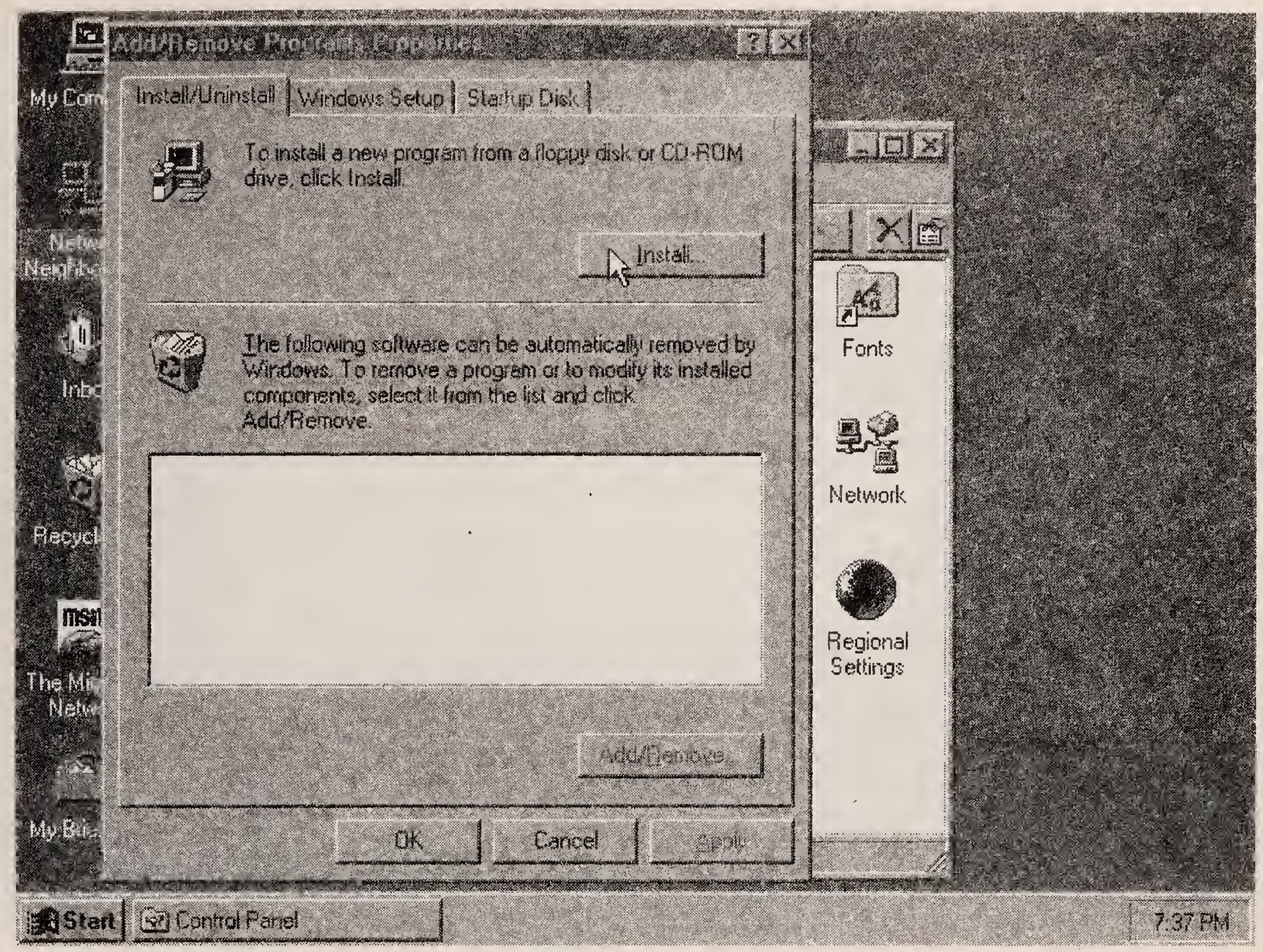
Open: a:\setup

图 1. 27　用 Run 对话框安装程序


My Computer
Network Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Control Panel
File
Edit
View
Help
Control Panel
Add New Hardware
Add/Remove Programs
Date/Time
Display
Fonts
Keyboard
Modems
Mouse
Multimedia
Network
Passwords
System
PC Card (PCMCIA)
Printers
Regional Settings
Sounds
Accessibility Options
Mail and Fax
Microsoft Mail Postoffice
Start
Control Panel
7:27 PM

Sets up programs and creates shortcuts.

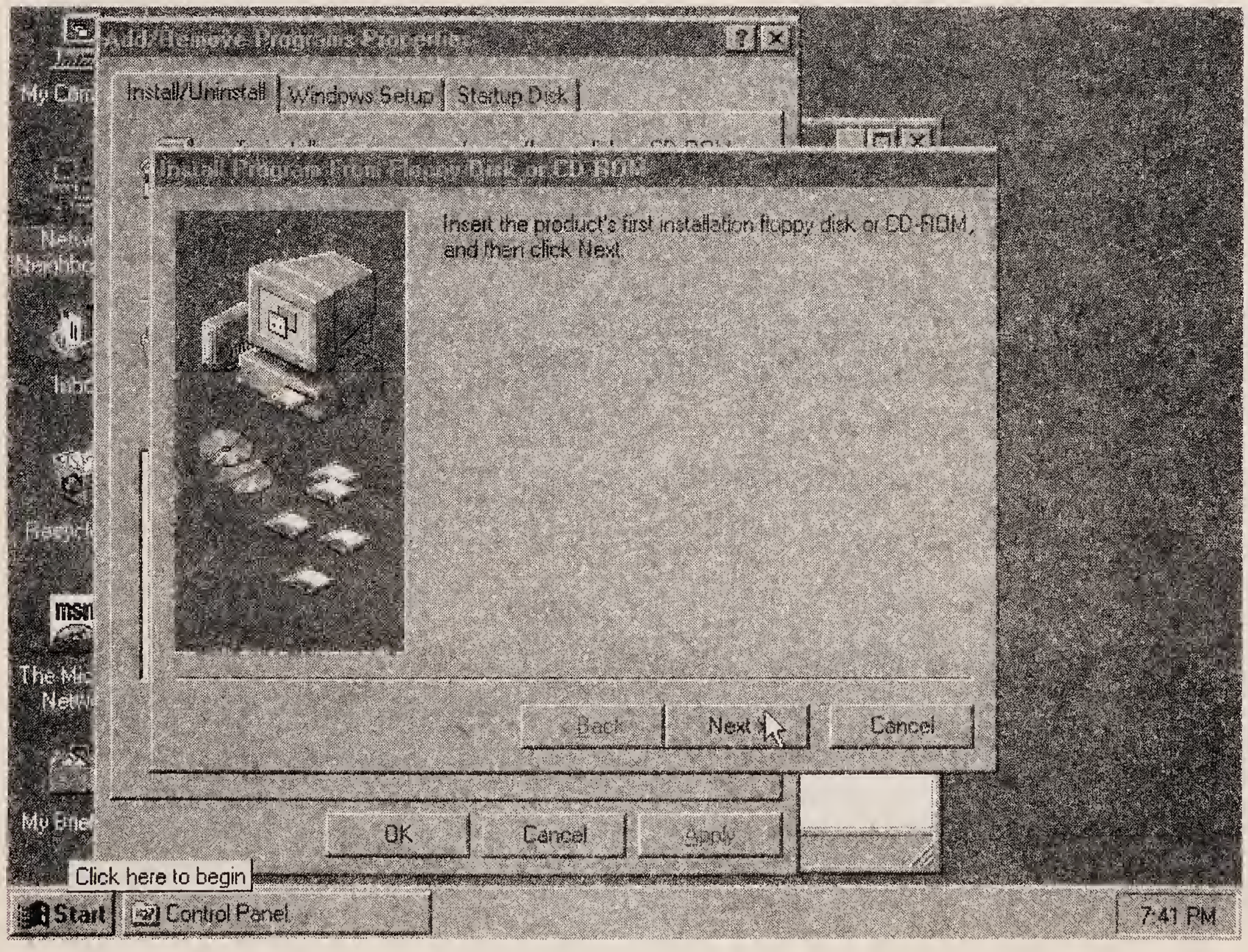
图 1. 28　控制面板(Control Panel)窗口


Add/Remove Programs Properties
Install/Uninstall
Windows Setup
Startup Disk
Install...
Add/Remove...
OK
Cancel
Apply
Fonts
Network
Regional Settings
My Com
Netw Neighbo
Inbo
Recycl
The Mi Netw
My Bri
Start
Control Panel
7:37 PM

To install a new program from a floppy disk or CD-ROM drive, click Install.

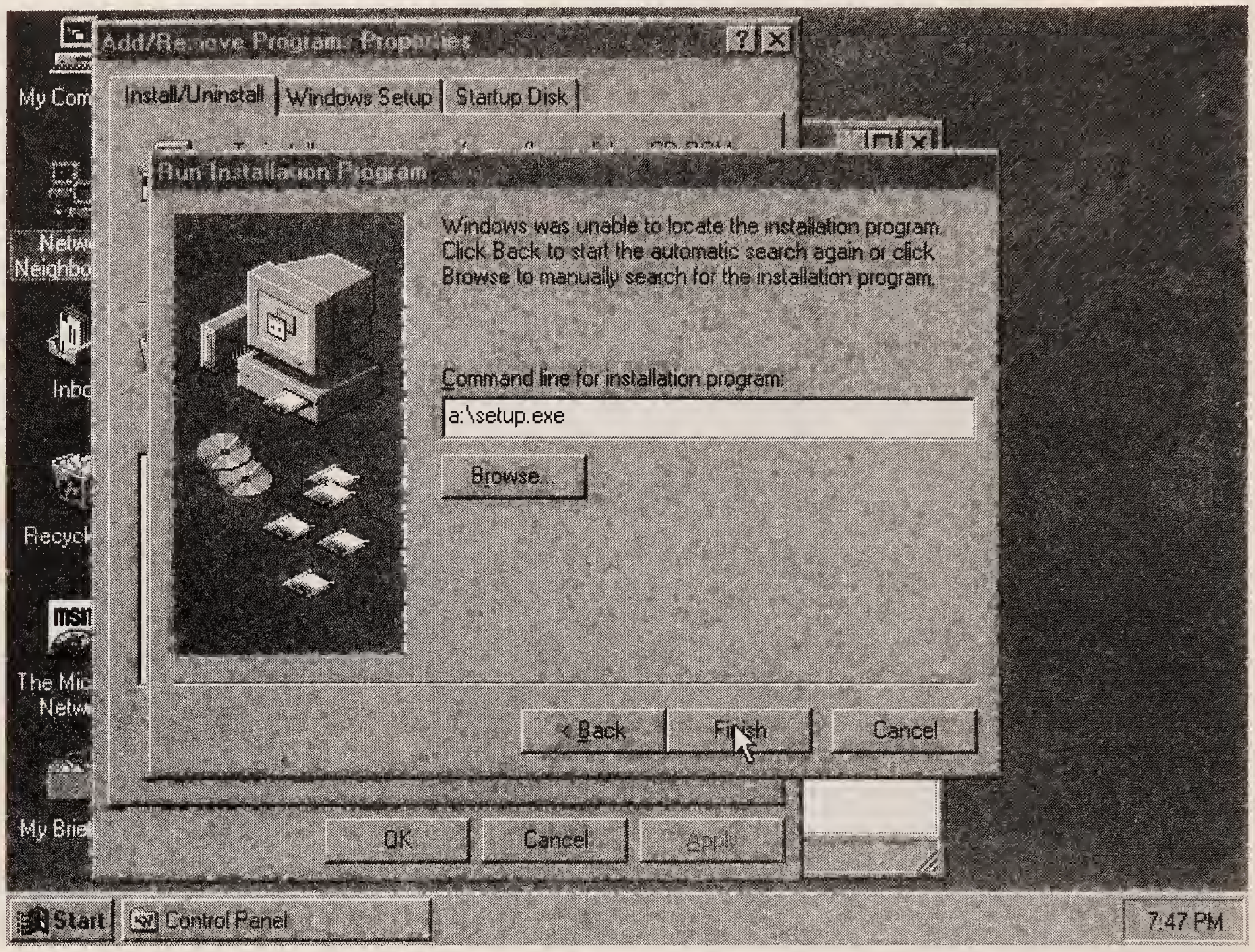
The following software can be automatically removed by Windows. To remove a program or to modify its installed components, select it from the list and click Add/Remove.

图 1. 29 Add/Remove Programs 对话框


Add/Remove Programs Properties
Install/Uninstall
Windows Setup
Startup Disk
Install Program From Floppy Disk or CD-ROM
< Back
Next >
Cancel
OK
Cancel
Apply
My Com
Netw Neighbo
Inbo
Recycl
The Mic Netw
My Brie
Click here to begin
Start
Control Panel
7:41 PM

Insert the product's first installation floppy disk or CD-ROM, and then click Next.

图 1. 30 提示插入含有安装程序的磁盘


Add/Remove Programs Properties
Install/Uninstall
Windows Setup
Startup Disk
Run Installation Program
Browse...
< Back
Finish
Cancel
OK
Cancel
Apply
My Com
Netw
Neighbo
Inbo
Recycl
The Mic
Netw
My Brie
Start
Control Panel
7:47 PM

Windows was unable to locate the installation program. Click Back to start the automatic search again or click Browse to manually search for the installation program.

Command line for installation program: a:\setup.exe

图 1.31　键入安装程序的路径及文件名

1.5　文件和文件夹的管理

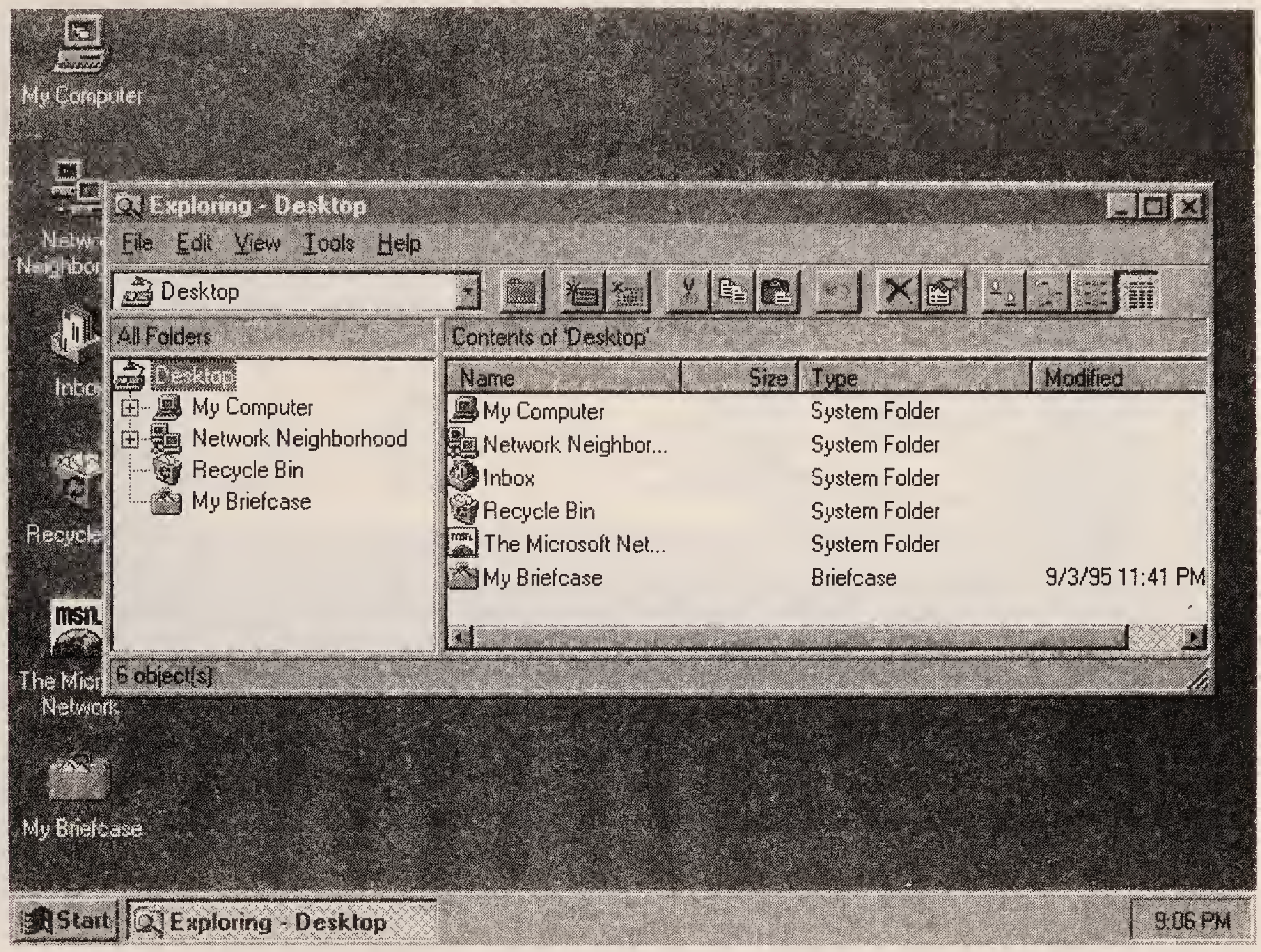
1.5.1　两个管理工具

Windows 95 有两个文件管理工具：My Computer 和 Explorer。对于大多数任务，用户既能用 My Computer 完成，也可以用 Explorer 来完成。并且两者的有些操作步骤也非常相似。但是，My Computer 没有利用 Windows 95 前期版本文件管理器所使用的层次和双区的方法。相反，Explorer 使用这些方法，并且充分利用了它们，创建了一个更强大的文件浏览工具，因此 Explorer 比 My Computer 更像 Windows 95 早期版本的文件管理器。

要想激活 Explorer，单击 Start 按钮，鼠标指向 Programs 项打开它的级联菜单，单击 Windows Explorer 选项，见图 1.32。

Explorer 窗口的左边是当前驱动器下所有文件夹列表，窗口的右部是当前文件夹的内容，它的显示内容随当前文件夹的不同而有所变化。当前文件夹的选择方法是在其图标上单击即可。例如，单击“My Computer”图标，屏幕变为图 1.33，可以看出它与图 1.32 有所不同，这是因为当前文件夹不同，在图 1.33 中，当前文件夹是“My Computer”。

在图 1.32 中用户可以看到，有些文件夹图标前有带有“＋”号的方框，这说明它还含有子文件夹，单击“＋”号，显示出它的子文件夹，并且“＋”号变成“－”号，见图 1.34，再单击“－”号可以关闭打开的子文件夹。

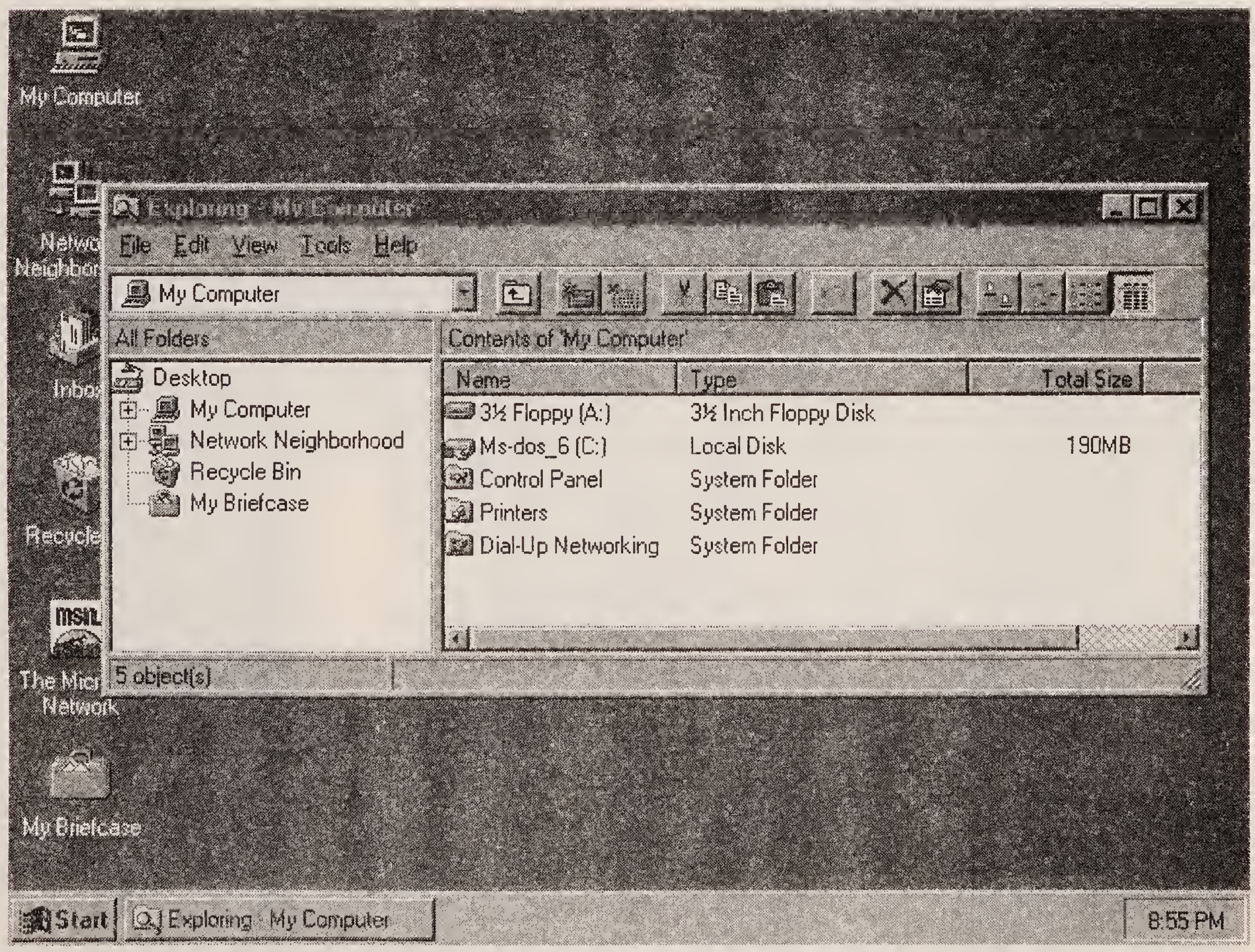

My Computer
Network Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Exploring - Desktop
File Edit View Tools Help
Desktop
All Folders
Desktop
My Computer
Network Neighborhood
Recycle Bin
My Briefcase
Start
Exploring - Desktop
9:06 PM

Contents of 'Desktop'

Name	Size	Type	Modified
My Computer		System Folder	
Network Neighbor...		System Folder	
Inbox		System Folder	
Recycle Bin		System Folder	
The Microsoft Net...		System Folder	
My Briefcase		Briefcase	9/3/95 11:41 PM

6 object(s)

图 1.32　激活 Explorer


My Computer
Network Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Exploring - My Computer
File Edit View Tools Help
My Computer
All Folders
Desktop
My Computer
Network Neighborhood
Recycle Bin
My Briefcase
Start
Exploring - My Computer
8:55 PM

Contents of 'My Computer'

Name	Type	Total Size
3½ Floppy (A:)	3½ Inch Floppy Disk	
Ms-dos_6 (C:)	Local Disk	190MB
Control Panel	System Folder	
Printers	System Folder	
Dial-Up Networking	System Folder	

5 object(s)

图 1.33　打开“My Computer 文件夹”

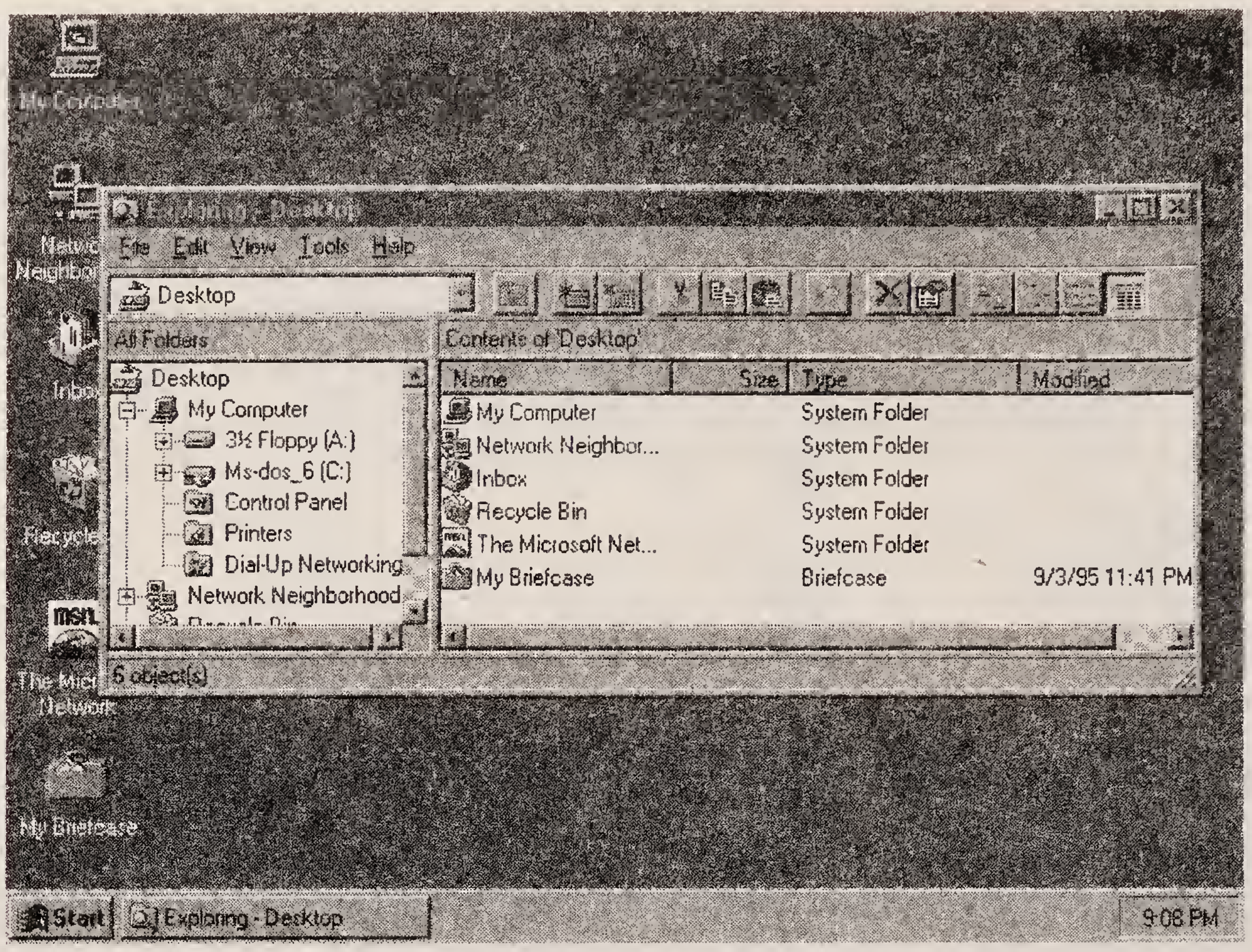

My Computer
Network Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Exploring - Desktop
File Edit View Tools Help
Desktop
All Folders
Contents of 'Desktop'
Desktop
My Computer
3½ Floppy (A:)
Ms-dos_6 (C:)
Control Panel
Printers
Dial-Up Networking
Network Neighborhood
Name
Size
Type
Modified
My Computer System Folder
Network Neighbor... System Folder
Inbox System Folder
Recycle Bin System Folder
The Microsoft Net... System Folder
My Briefcase Briefcase 9/3/95 11:41 PM
6 object(s)
Start
Exploring - Desktop
9:08 PM

图 1.34 打开子文件夹

1.5.2 文件夹的操作

(1) 文件夹的创建、删除和更名

在 My Computer 中，打开用户想存放新文件夹的驱动器或文件夹(双击图标即可)。例如，双击“Ms-dos _ 6 C:”的图标，然后把鼠标指向 File 菜单的 New 选项以便打开它的级联菜单，再单击级联菜单的“Folder”命令，见图 1.35。这时出现一个带有缺省名为 New Folder 的新文件夹，用户可以在它下面输入新文件的名字，然后回车即可，见图 1.36。

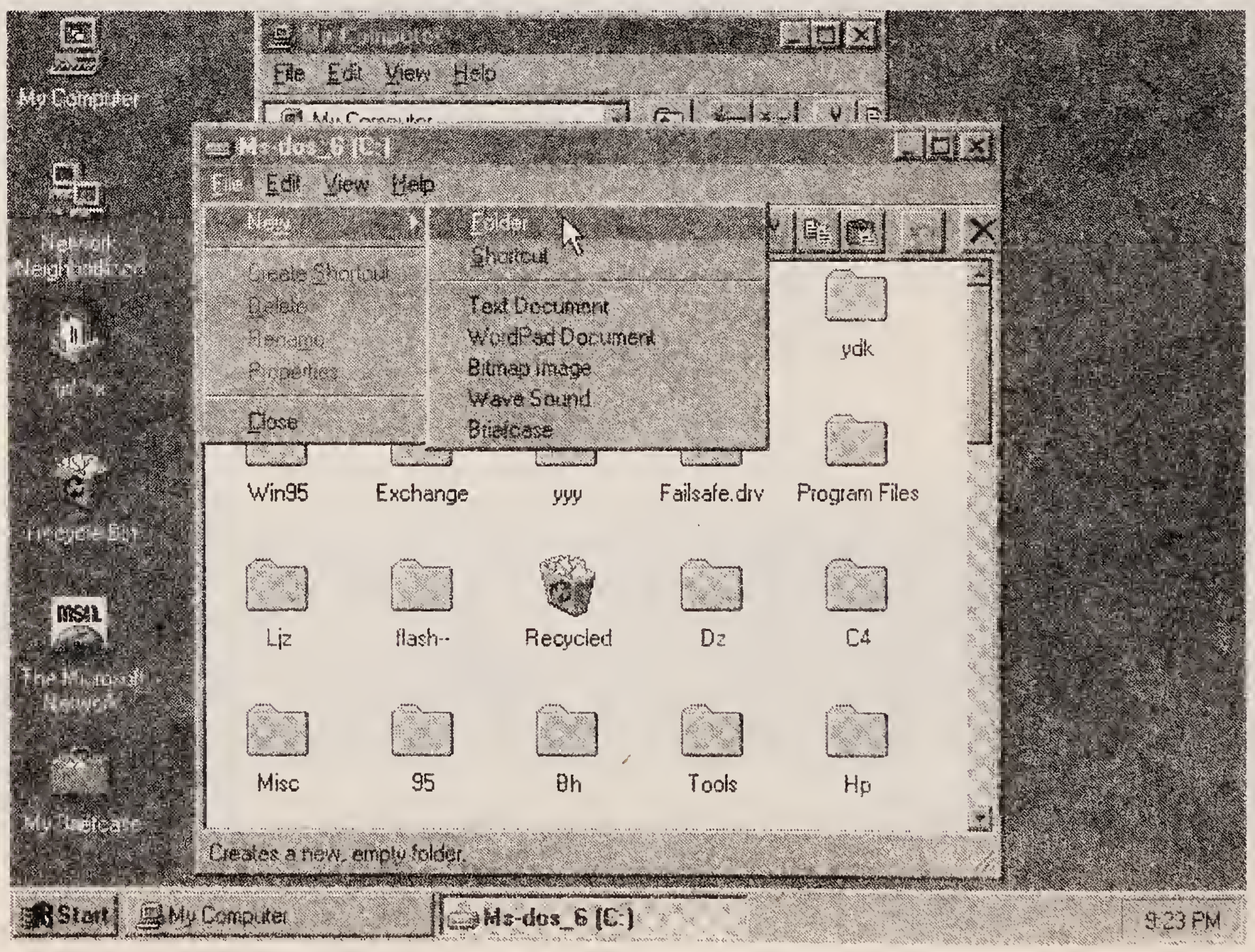

My Computer
Network Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
My Computer
File Edit View Help
Ms-dos_6 (C:)
File Edit View Help
New
Create Shortcut
Delete
Rename
Properties
Close
Folder
Shortcut
Text Document
WordPad Document
Bitmap Image
Wave Sound
Briefcase
ydk
Win95
Exchange
yyy
Failsafe.drv
Program Files
Liz
flash--
Recycled
Dz
C4
Misc
95
Bh
Tools
Hp
Creates a new, empty folder.
Start
My Computer
Ms-dos_6 (C:)
9:23 PM

图 1.35 建立新文件夹的菜单命令

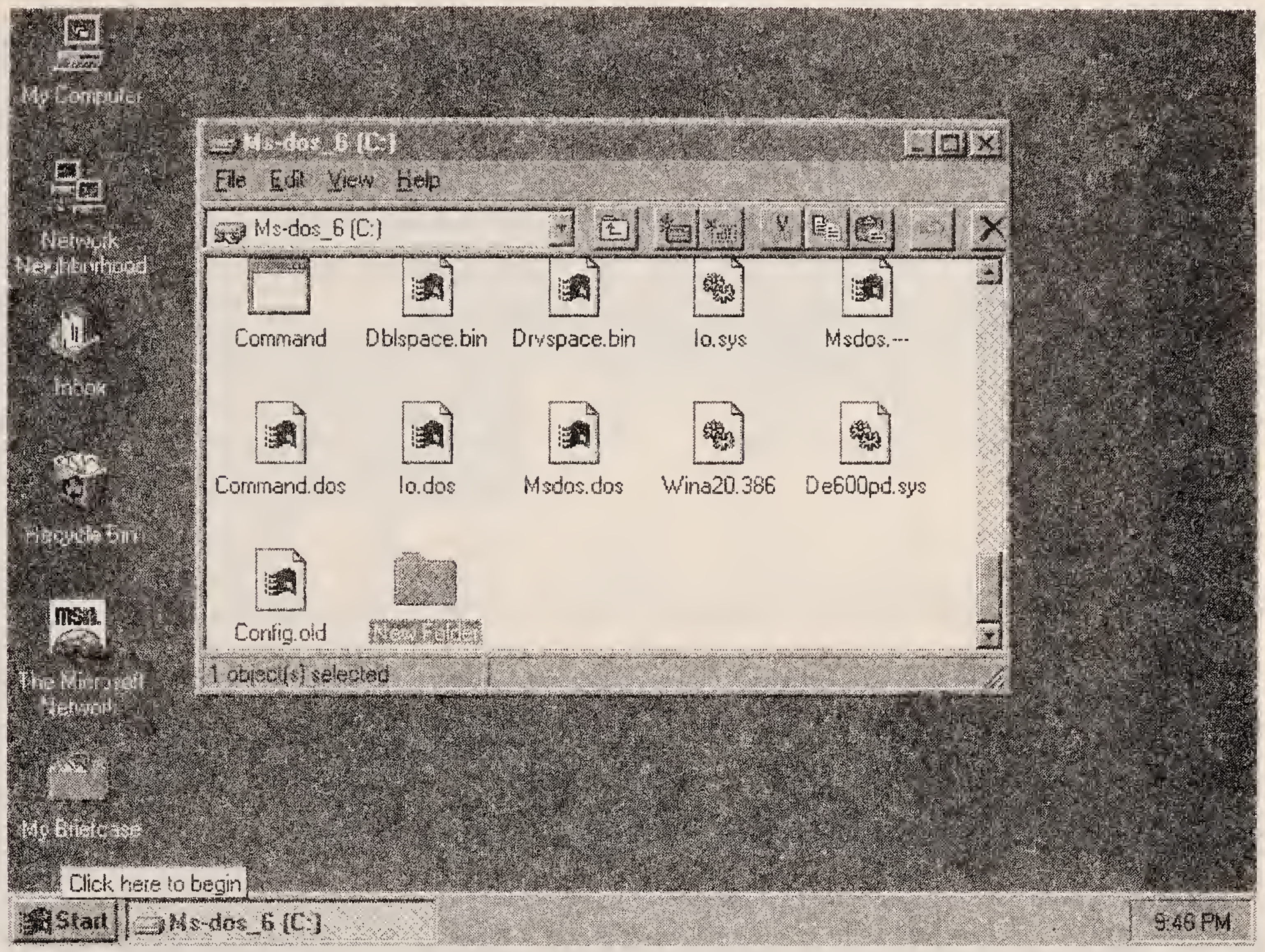

My Computer
Network Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Ms-dos_6 (C:)
File Edit View Help
Ms-dos_6 (C:)
Command
Dblspace.bin
Drvspace.bin
Io.sys
Msdos.---
Command.dos
Io.dos
Msdos.dos
Wina20.386
De600pd.sys
Config.old
New Folder
1 object(s) selected
Click here to begin
Start
Ms-dos_6 (C:)
9:46 PM

图 1.36 建立的新文件夹

要删除一个文件夹，用鼠标右按钮单击用户想要删除的文件夹，这时出现一个菜单，单击菜单上的 Delete 命令就完成文件夹的删除工作，见图 1.37。

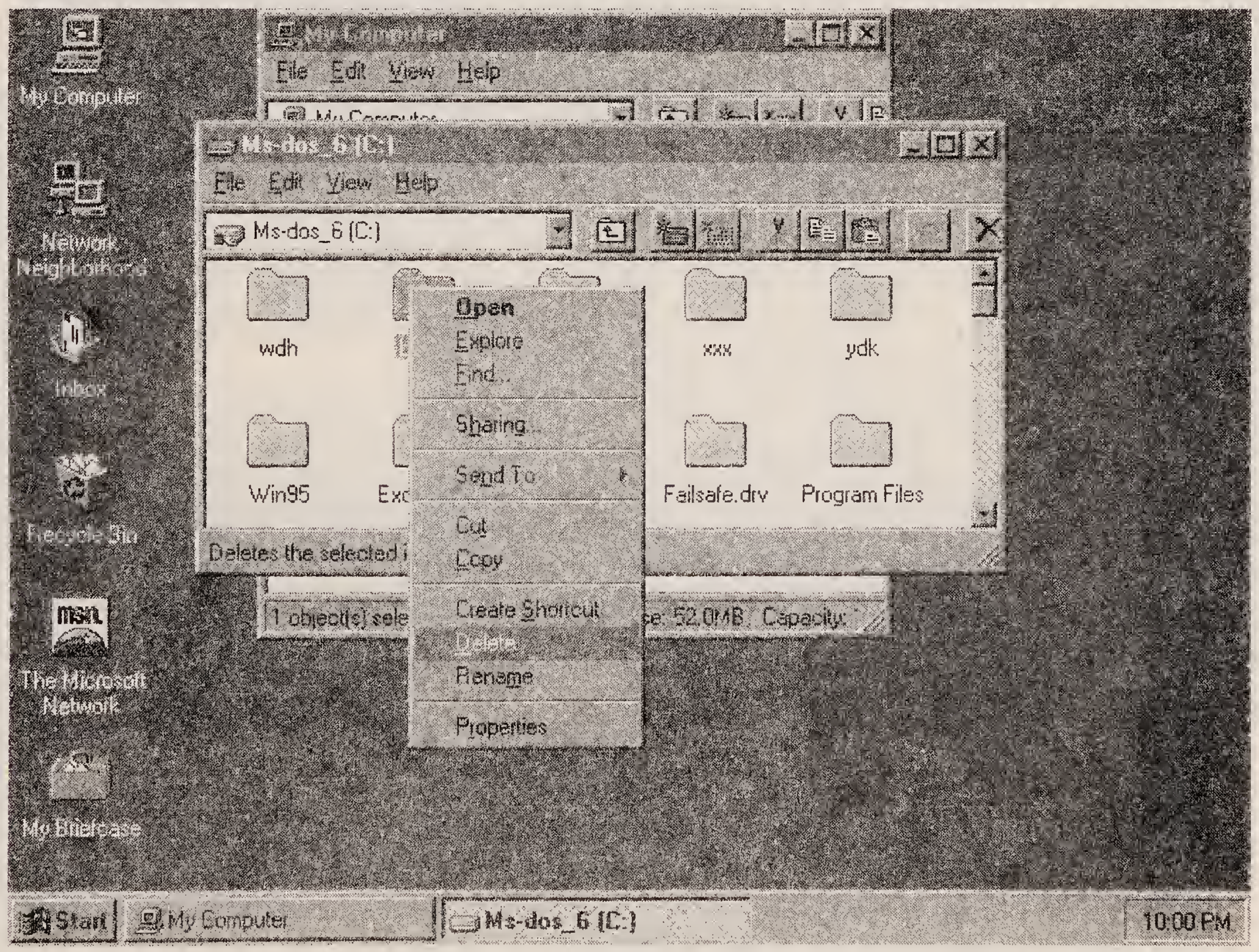

My Computer
Network Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
My Computer
File Edit View Help
Ms-dos_6 (C:)
File Edit View Help
Ms-dos_6 (C:)
wdh
xxx
ydk
Win95
Exc
Failsafe.drv
Program Files
Open
Explore
Find...
Sharing...
Send To
Cut
Copy
Create Shortcut
Delete
Rename
Properties
Deletes the selected i
1 object(s) sele
ce: 52.0MB Capacity:
Start
My Computer
Ms-dos_6 (C:)
10:00 PM

图 1.37 删除一个文件夹

更改一个文件夹名字的方法和删除操作相似，在图 1.37 的菜单上单击 Rename 命令即可。

(2) 文件夹的拷贝和移动

用户如果想拷贝或移动文件夹可以按下面步骤操作：

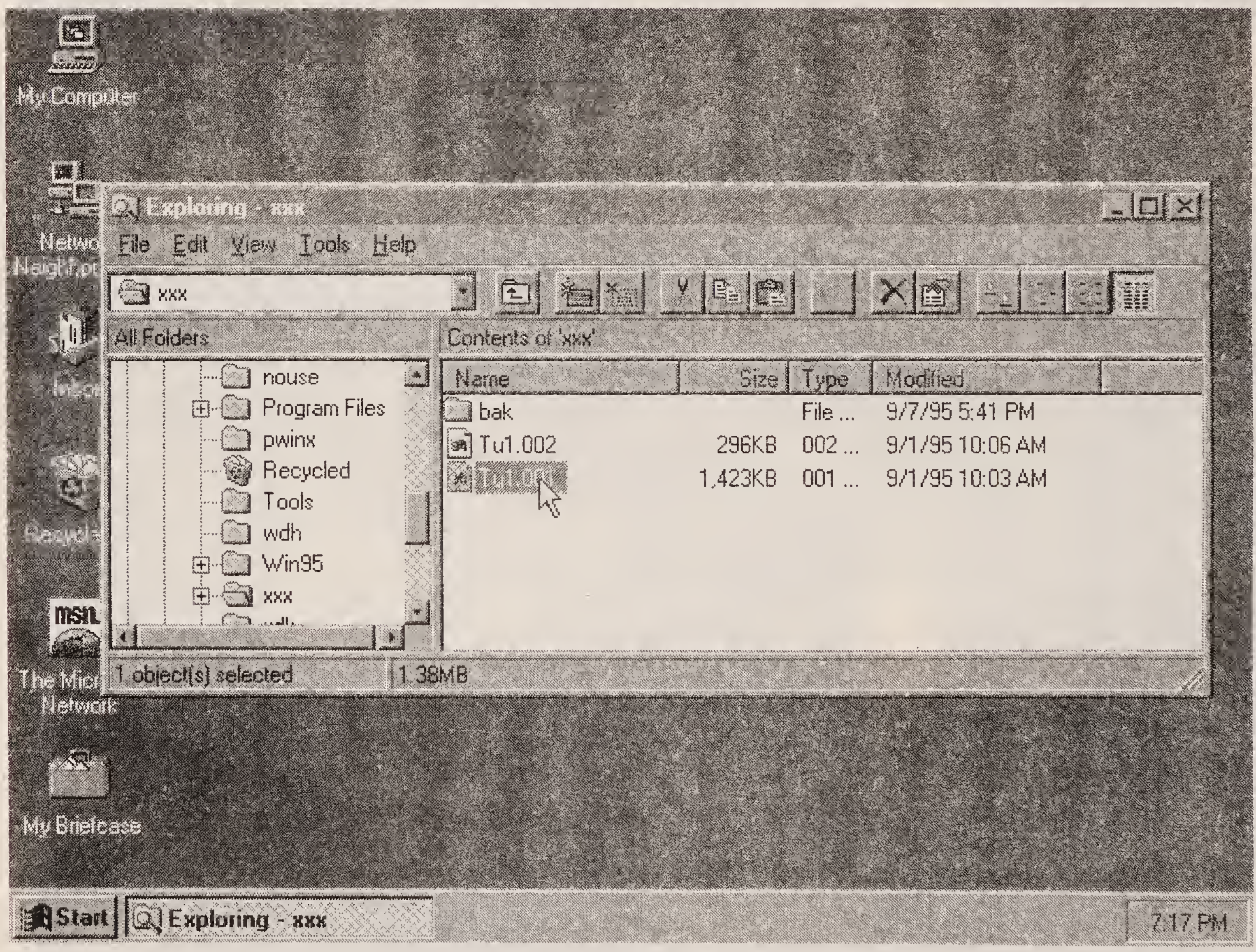
1) 在 My Computer 或 Explorer 中选择用户想拷贝或移动的文件夹。
2) 打开 Edit 菜单，单击 Copy(拷贝)或 Move(移动)命令。
3) 打开用户想存放文件夹的驱动器或文件夹。
4) 打开 Edit 菜单，单击 Paste 命令。

1.5.3 文件操作

用 My Computer 或资源管理器(Windows Explorer)可方便地进行文件的有关操作。

(1) 文件的移动和拷贝

打开 Explorer 窗口(单击图标)，选择要想移动或拷贝的文件，见图 1.38。然后按住鼠标左按钮拖动图标到新的驱动器或文件夹中，就可以完成移动操作。用 My Computer，以同样的方法也可完成文件的移动。拷贝文件同移动文件操作类似，但在拖动图标的同时要按住键盘上 Ctrl 键。


My Computer
Network Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Exploring - xxx
File Edit View Tools Help
xxx
All Folders
Contents of 'xxx'
nouse
Program Files
pwinx
Recycled
Tools
wdh
Win95
xxx
1 object(s) selected
1.38MB
Start
Exploring - xxx
7:17 PM

Name	Size	Type	Modified
bak		File ...	9/7/95 5:41 PM
Tu1.002	296KB	002 ...	9/1/95 10:06 AM
Tu1.001	1,423KB	001 ...	9/1/95 10:03 AM

图 1.38 选择文件

有时需要向软盘上拷贝文件，如果想做此项工作，首先要选择需拷贝的文件，接着打开 File 菜单，把鼠标指向“Send to”选项以打开它的子菜单，然后单击要想拷贝到的驱动器图标，至此就可以向软驱上拷贝文件的工作，见图 1.39。

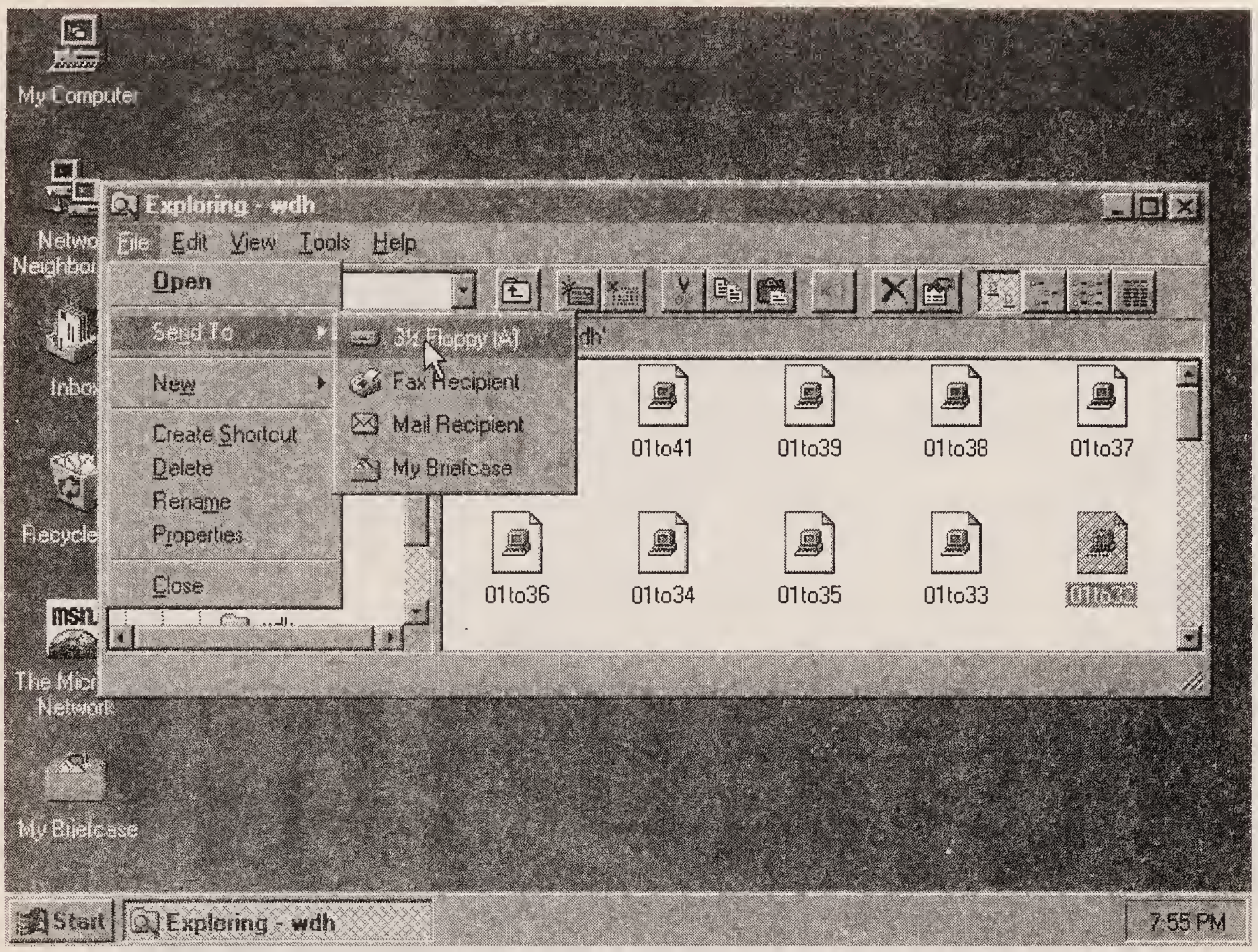

My Computer
Network Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Exploring - wdh
File Edit View Tools Help
Open
Send To
New
Create Shortcut
Delete
Rename
Properties
Close
3½ Floppy (A)
Fax Recipient
Mail Recipient
My Briefcase
01to41
01to39
01to38
01to37
01to36
01to34
01to35
01to33
01to32
Start
Exploring - wdh
7:55 PM

图 1.39 向软盘上拷贝文件

有时想要同时移动或拷贝多个文件，选取多个文件的方法是：按住键盘上的 Ctrl 键的同时依次单击所想选择的文件，见图 1.40。选择多个连续的文件可按下 Shift 键，单击头尾两个文件即可。

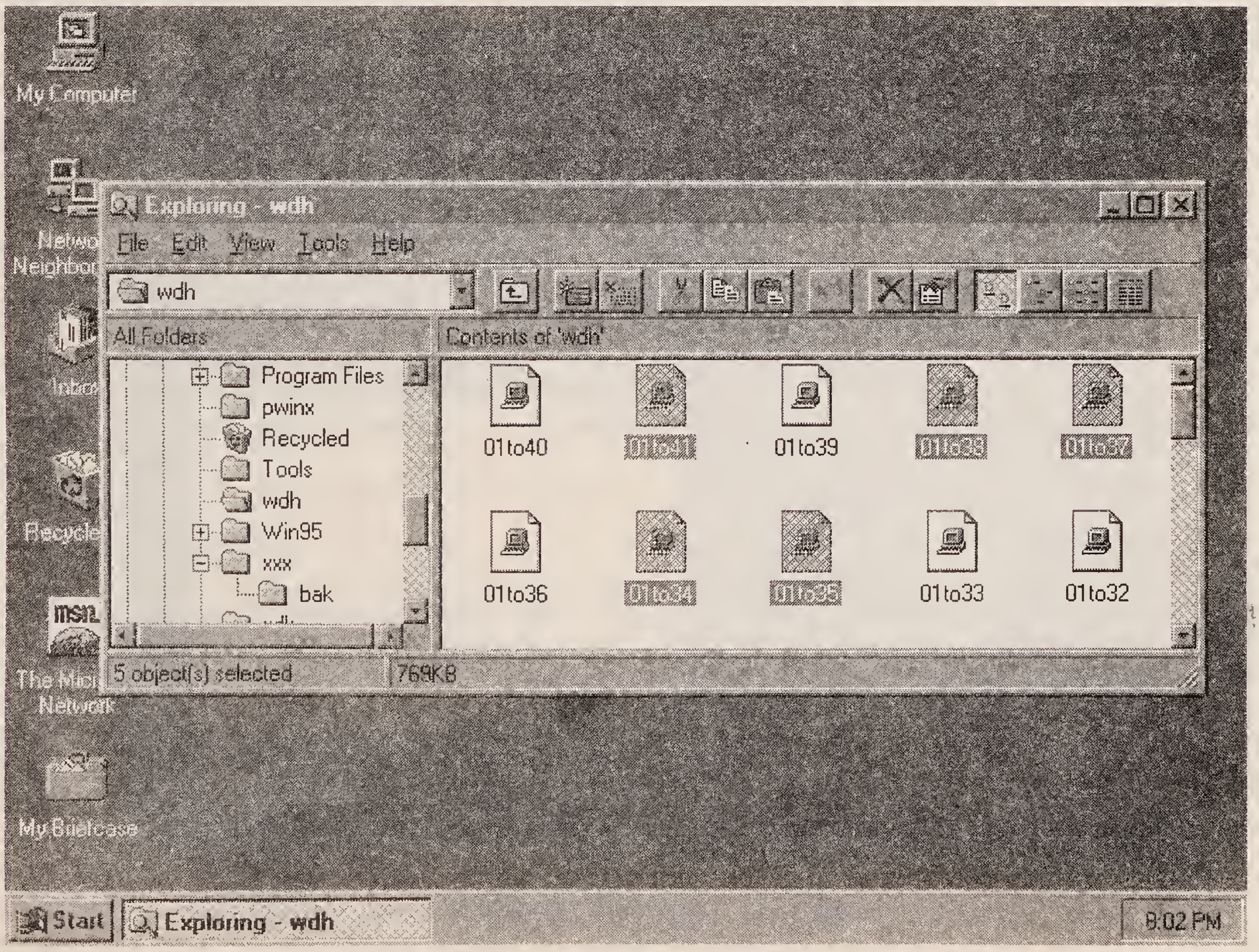

My Computer
Network Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Exploring - wdh
File Edit View Tools Help
wdh
All Folders
Contents of 'wdh'
Program Files
pwinx
Recycled
Tools
wdh
Win95
xxx
bak
01to40
01to41
01to39
01to38
01to37
01to36
01to34
01to35
01to33
01to32
5 object(s) selected
769KB
Start
Exploring - wdh
8:02 PM

图 1.40 同时选择多个文件

（2）文件的删除和更名

文件的删除和更名最快捷的方法是用鼠标右按钮单击文件图标，这时出现一菜单，单击“Delete”或“Rename”就可以删除文件或更改文件名，见图 1.41。单击“File”菜单中的“Delete”或“Rename”也可以删除文件或更改文件名。

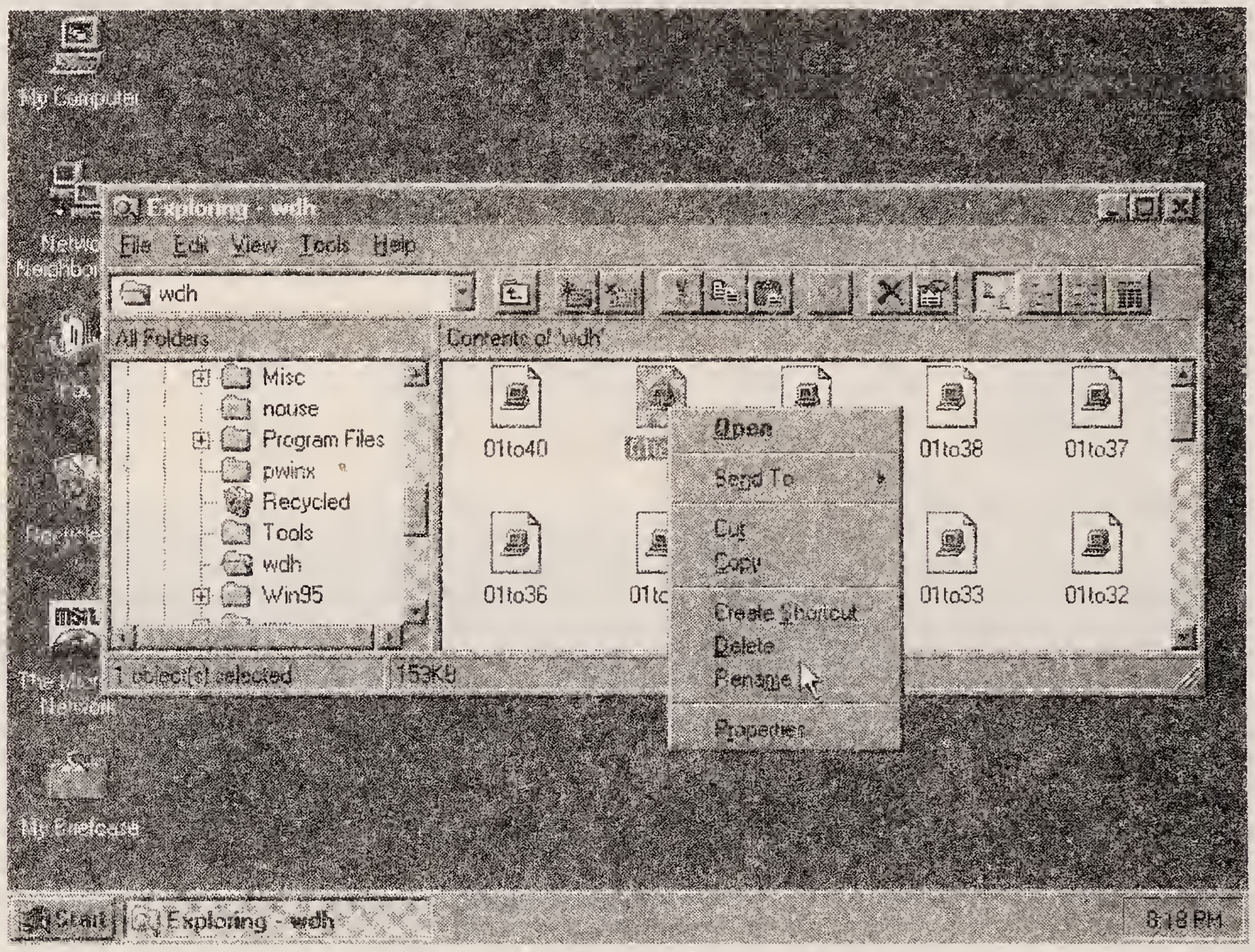

My Computer
Network Neighborhood
The Microsoft Network
My Briefcase
Exploring - wdh
File Edit View Tools Help
wdh
All Folders
Contents of 'wdh'
Misc
nouse
Program Files
pwinx
Recycled
Tools
wdh
Win95
01to40
01to38
01to37
01to36
01to33
01to32
Open
Send To
Cut
Copy
Create Shortcut
Delete
Rename
Properties
1 object(s) selected
153KB
Start
Exploring - wdh
8:18 PM

图 1.41 利用鼠标右按钮删除和更名文件

（3）文件的打开和保存

单击“File”菜单中的“Open”或“Save”、“Save as...”可以打开或保存一个文件，见图 1.42。另外，用 Start 菜单可以打开近期被打开的文件，见图 1.43。

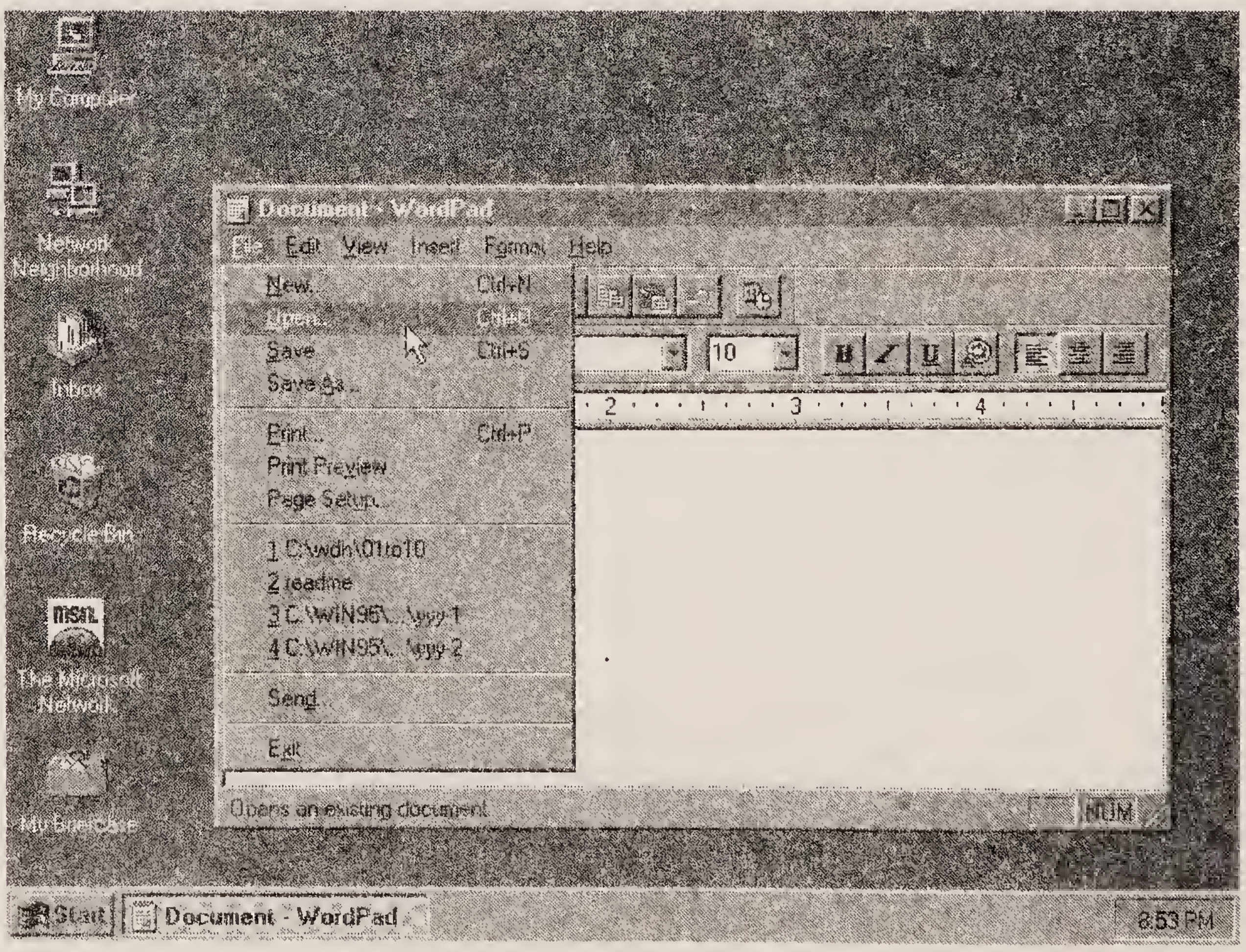

My Computer
Network Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Document - WordPad
File Edit View Insert Format Help
New... Ctrl+N
Open... Ctrl+O
Save Ctrl+S
Save As...
Print... Ctrl+P
Print Preview
Page Setup...
1 C:\wdh\01to10
2 readme
3 C:\WIN95\...\yyy-1
4 C:\WIN95\...\yyy-2
Send...
Exit
10
B I U
Opens an existing document
NUM
Start
Document - WordPad
8:53 PM

图 1.42 文件的打开或保存

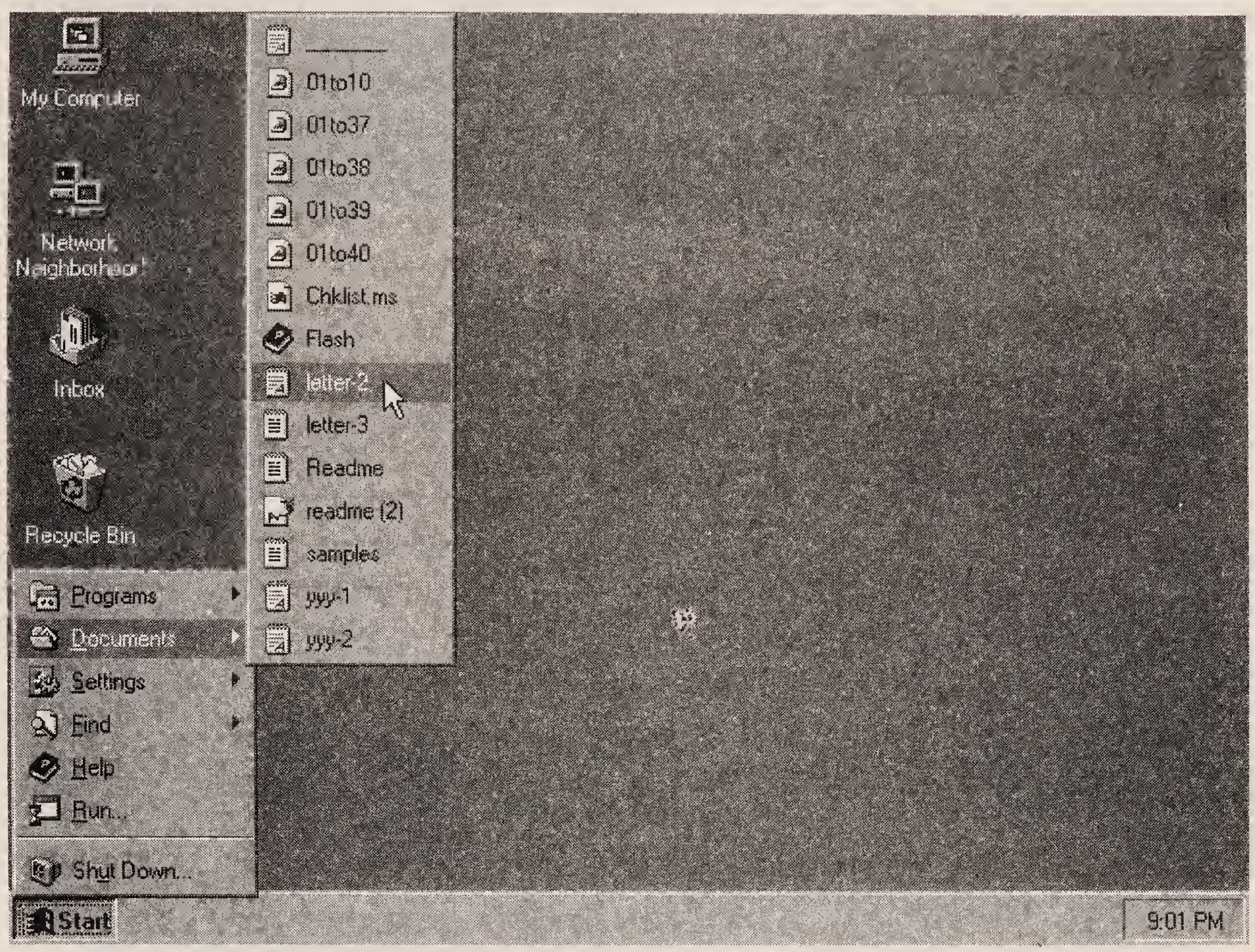

My Computer
Network Neighborhood
Inbox
Recycle Bin
01to10
01to37
01to38
01to39
01to40
Chklist.ms
Flash
letter-2
letter-3
Readme
readme (2)
samples
yyy-1
yyy-2
Programs
Documents
Settings
Find
Help
Run...
Shut Down...
Start
9:01 PM

图 1.43　打开近期被打开的文件

1.6　Windows 95 的帮助系统及查找功能

Windows 95 提供了强大的帮助系统，利用它是学习和使用 Windows 95 的最佳途径。它对 Windows 早期版本的联机帮助系统做了许多改进，使用起来更为方便。

帮助系统的打开可以通过 Start 菜单的“Help”选项。也可以用某个应用程序菜单条上的“Help”菜单获取帮助信息。

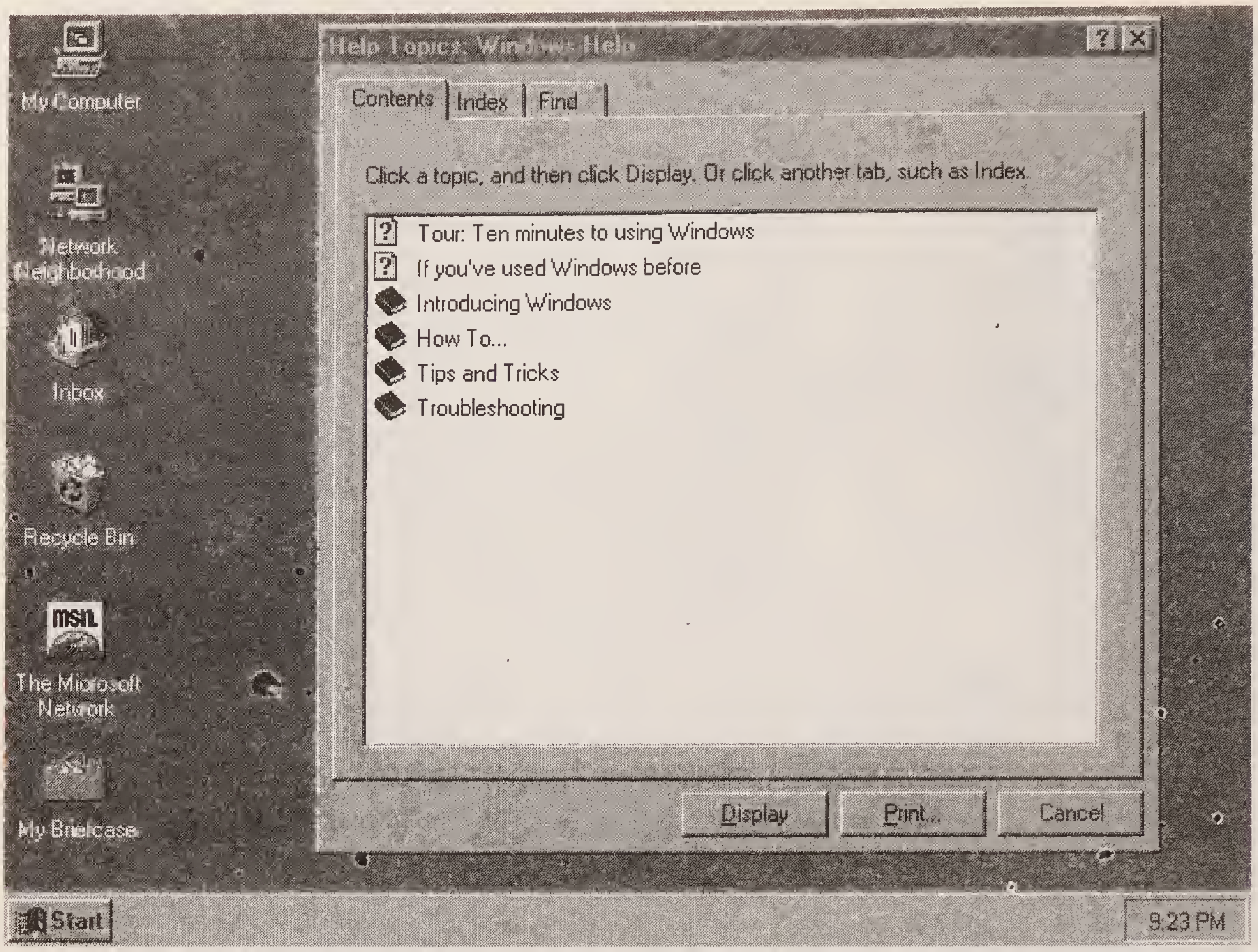
1.6.1　帮助窗口的基本使用方法

单击 Start 菜单“Help”图标，就可以打开帮助窗口，见图 1.44。

从图中可以看到窗口顶部有三个标记：“Contents”、“Index”、“Find”。其中“Contents”表示目录帮助形式，“Index”表示索引帮助形式，“Find”表示查找帮助形式。

(1) 目录帮助形式

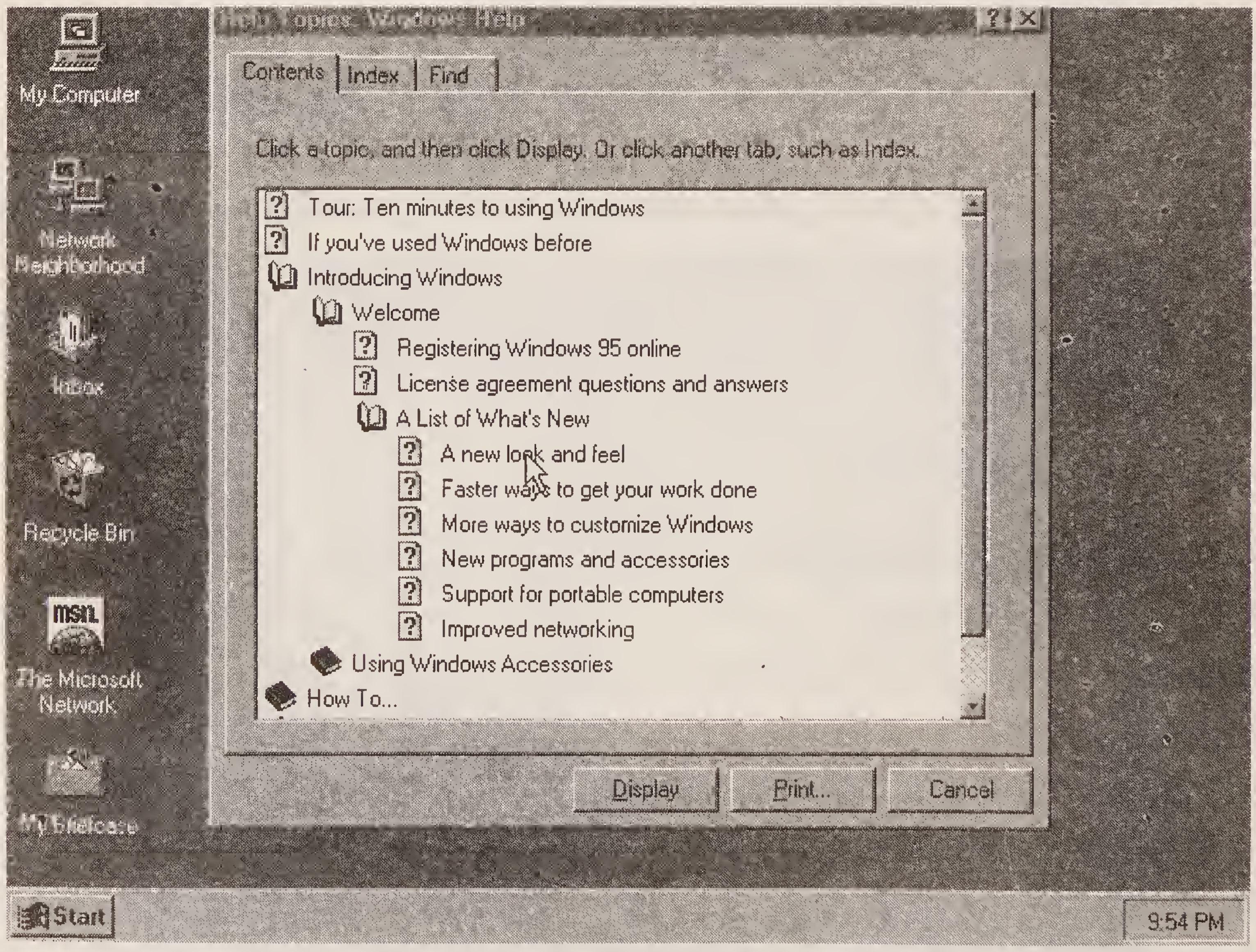
单击“Contents”就显示目录帮助形式的窗口，见图 1.45。由图中可以看出，该形式以目录方式安排帮助信息，目录以书图标表示，单击书图标，再单击“Open”命令按钮就可以打开一级目录，此时书图标也被打开，好像打开一本书一样，也可以通过双击打开一级目录。重复上述操作直到“?”图标出现为止，见图 1.45。通过双击被打开的书图标可关闭一级目录。


My Computer
Network Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Help Topics: Windows Help
Contents
Index
Find
Display
Print...
Cancel
Start
9:23 PM

Click a topic, and then click Display. Or click another tab, such as Index.

- Tour: Ten minutes to using Windows
- If you've used Windows before
- Introducing Windows
- How To...
- Tips and Tricks
- Troubleshooting

图 1.44 打开帮助窗口

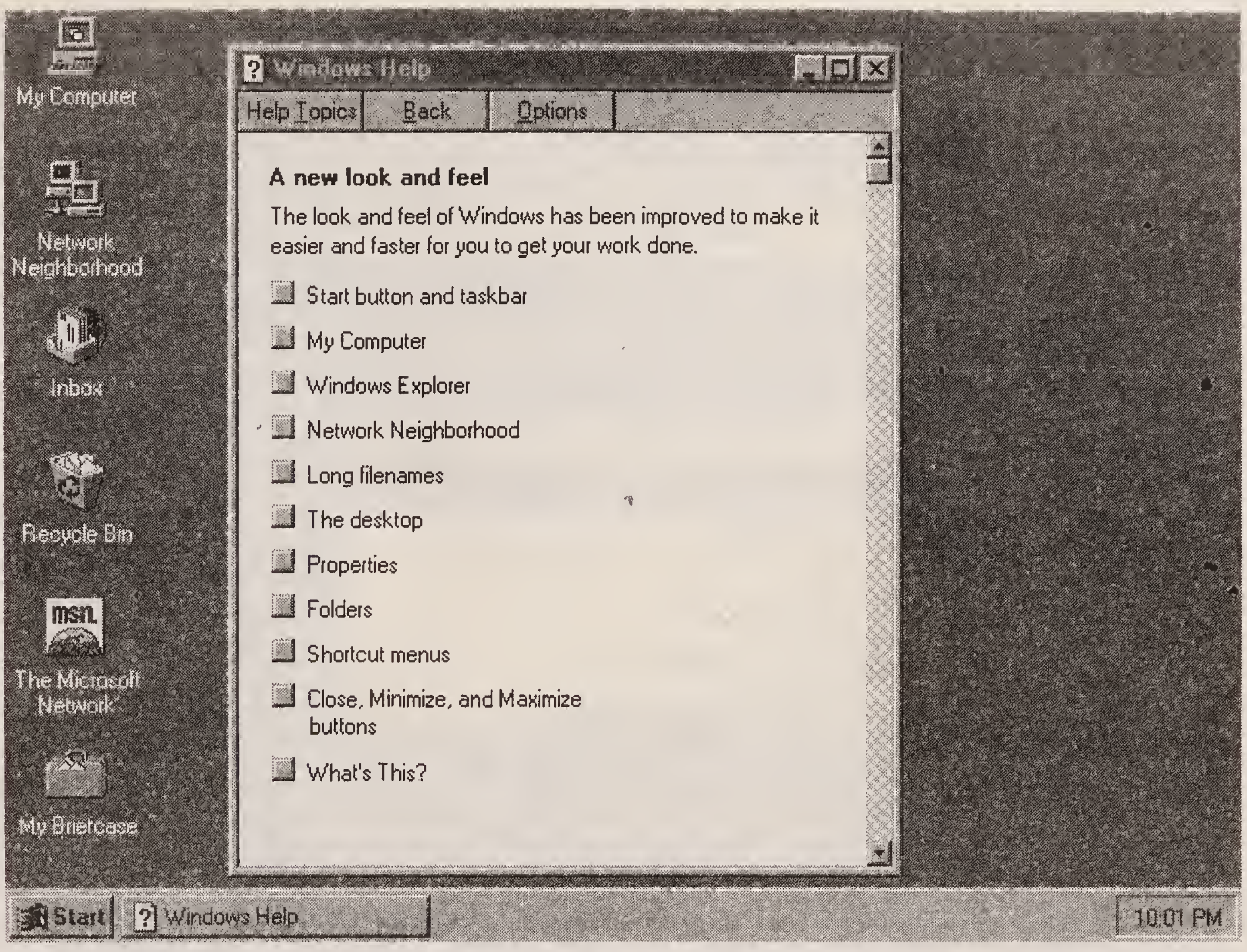

My Computer
Network Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Help Topics: Windows Help
Contents
Index
Find
Display
Print...
Cancel
Start
9:54 PM

Click a topic, and then click Display. Or click another tab, such as Index.

- Tour: Ten minutes to using Windows
- If you've used Windows before
- Introducing Windows
  - Welcome
    - Registering Windows 95 online
    - License agreement questions and answers
    - A List of What's New
      - A new look and feel
      - Faster ways to get your work done
      - More ways to customize Windows
      - New programs and accessories
      - Support for portable computers
      - Improved networking
  - Using Windows Accessories
- How To...

图 1.45 目录帮助形式

双击某一项的“？”图标就可以显示有关帮助信息。也可以单击“？”图标，再单击“Display”按钮。图 1.46 是打开某一项的显示结果。


My Computer
Network Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Windows Help
Help Topics
Back
Options
Start
Windows Help
10:01 PM

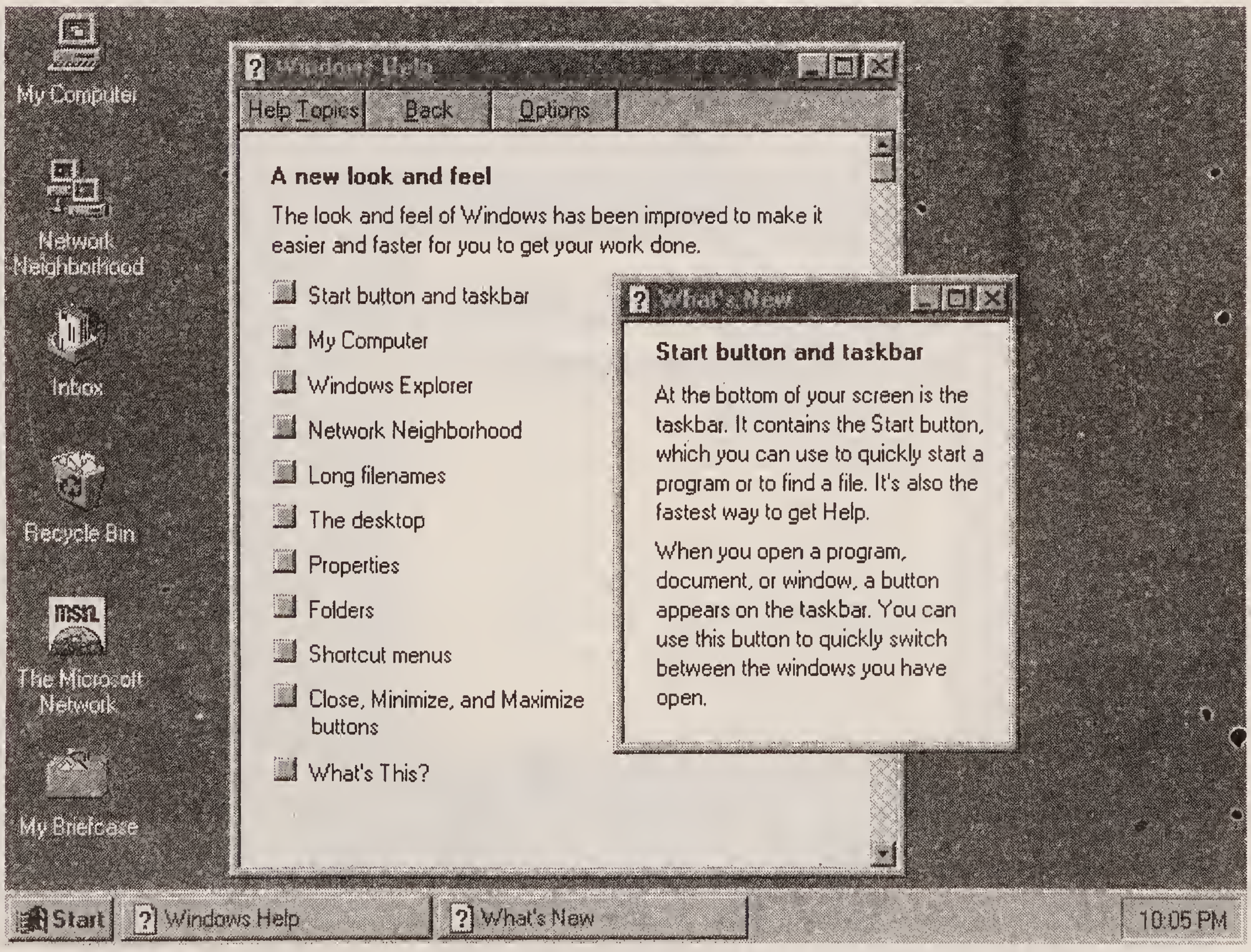
A new look and feel

The look and feel of Windows has been improved to make it easier and faster for you to get your work done.

- Start button and taskbar
- My Computer
- Windows Explorer
- Network Neighborhood
- Long filenames
- The desktop
- Properties
- Folders
- Shortcut menus
- Close, Minimize, and Maximize buttons
- What's This?

图 1.46 帮助信息的帮助主题

图 1.46 中，显示出许多有关的帮助主题，单击某一项，就得到该项的具体帮助信息，见图 1.47。


My Computer
Network Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Windows Help
Help Topics
Back
Options
What's New
Start
Windows Help
What's New
10:05 PM

A new look and feel

The look and feel of Windows has been improved to make it easier and faster for you to get your work done.

- Start button and taskbar
- My Computer
- Windows Explorer
- Network Neighborhood
- Long filenames
- The desktop
- Properties
- Folders
- Shortcut menus
- Close, Minimize, and Maximize buttons
- What's This?

Start button and taskbar

At the bottom of your screen is the taskbar. It contains the Start button, which you can use to quickly start a program or to find a file. It's also the fastest way to get Help.

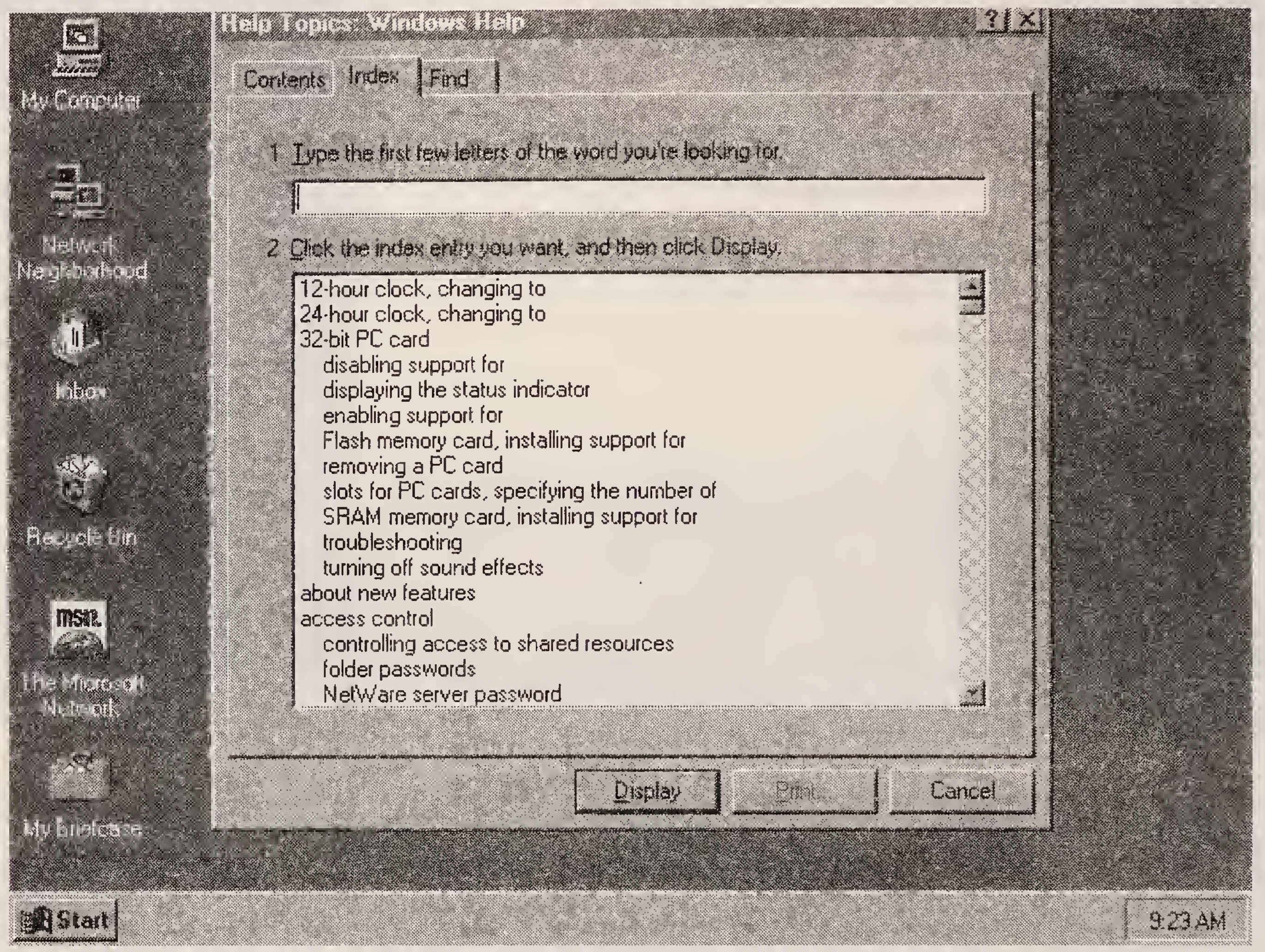
When you open a program, document, or window, a button appears on the taskbar. You can use this button to quickly switch between the windows you have open.

图 1.47 具体项的帮助信息

想返回帮助主题，单击“Help Topics”，想回到上一次帮助信息单击“Back”按钮。

（2）索引帮助形式

单击“Index”切换到索引帮助形式，见图 1.48。在第一个域中，键入一个关键词——就是一个描述用户所寻求帮助题目的词。例如用户键入一个关键词 Copy，以获取有关拷贝的信息，随着键入关键词，用户注意到在用户只键入三个字母时，Help 就查到了题目（在这个例子中，是键入“Cop”）。这是因为 Help 在每键入一个字符时都在搜索，见图 1.49。

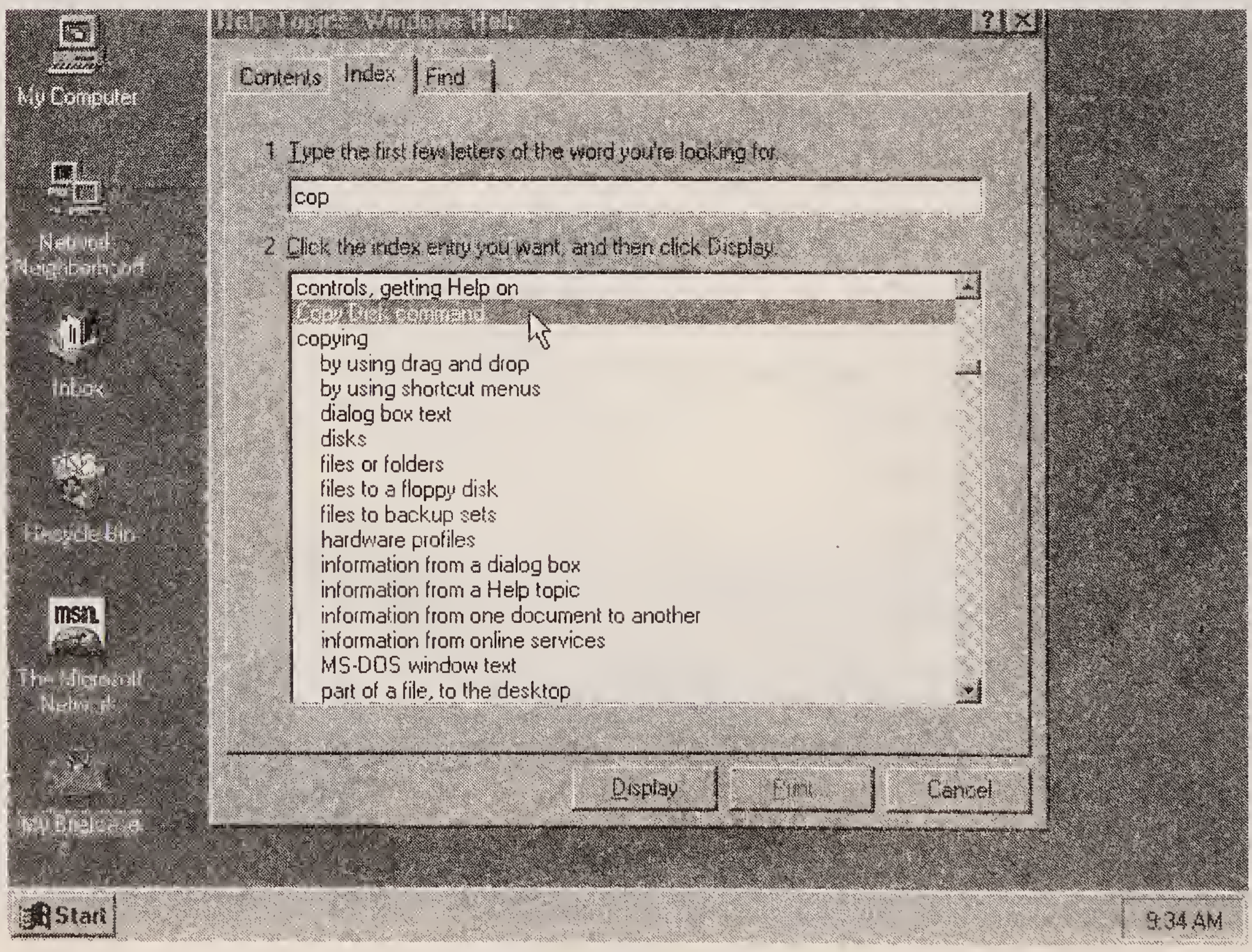

Help Topics: Windows Help
?
×
Contents
Index
Find
Display
Print...
Cancel
My Computer
Network Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Start
9:23 AM

1 Type the first few letters of the word you're looking for.

2 Click the index entry you want, and then click Display.

12-hour clock, changing to
24-hour clock, changing to
32-bit PC card
  disabling support for
  displaying the status indicator
  enabling support for
  Flash memory card, installing support for
  removing a PC card
  slots for PC cards, specifying the number of
  SRAM memory card, installing support for
  troubleshooting
  turning off sound effects
about new features
access control
  controlling access to shared resources
  folder passwords
  NetWare server password

图 1.48　索引帮助窗口


Help Topics: Windows Help
?
×
Contents
Index
Find
Display
Print...
Cancel
My Computer
Network Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Start
9:34 AM

1 Type the first few letters of the word you're looking for.

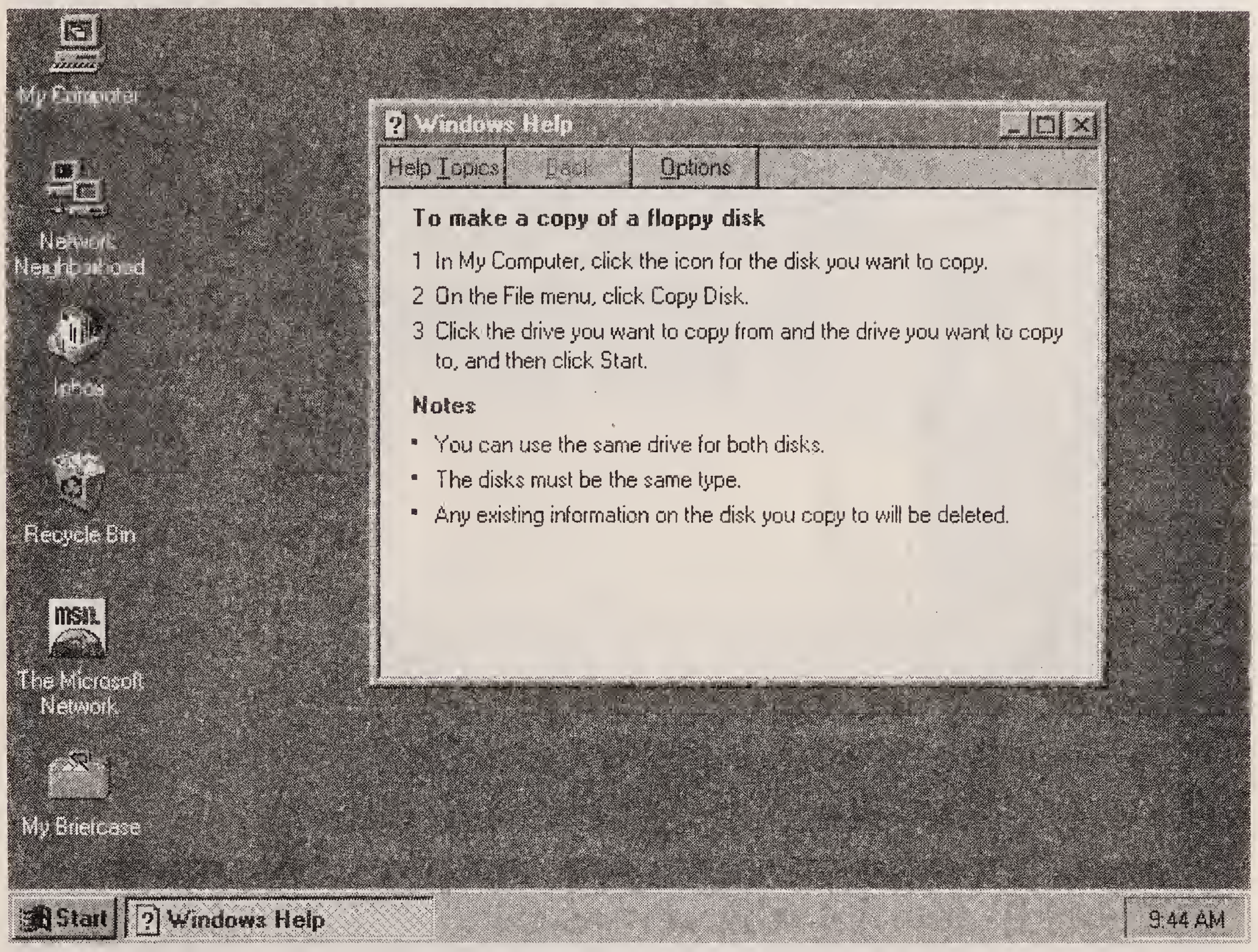
cop

2 Click the index entry you want, and then click Display.

controls, getting Help on
Copy Disk command
copying
  by using drag and drop
  by using shortcut menus
  dialog box text
  disks
  files or folders
  files to a floppy disk
  files to backup sets
  hardware profiles
  information from a dialog box
  information from a Help topic
  information from one document to another
  information from online services
  MS-DOS window text
  part of a file, to the desktop

图 1.49　键入关键词

双击有关帮助题目，或者单击题目，再单击“Display”按钮，就可获取有关帮助信息。例如，双击图 1.49 中的“Copy Disk Command 就可以获取有关信息，见图 1.50。


My Computer
Network Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Windows Help
Help Topics
Back
Options
Start
Windows Help
9:44 AM

To make a copy of a floppy disk

1 In My Computer, click the icon for the disk you want to copy.

2 On the File menu, click Copy Disk.

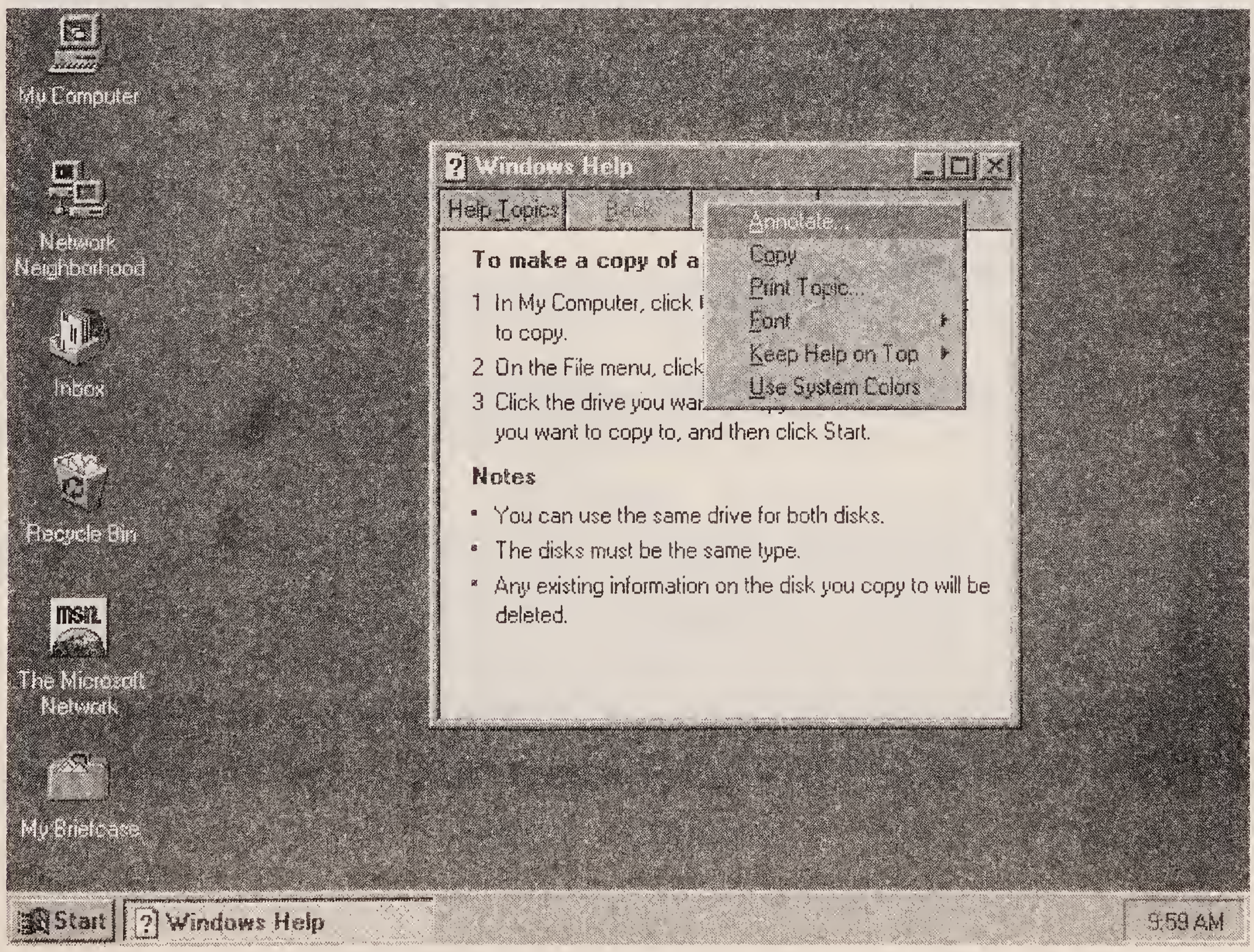
3 Click the drive you want to copy from and the drive you want to copy to, and then click Start.

Notes

- You can use the same drive for both disks.
- The disks must be the same type.
- Any existing information on the disk you copy to will be deleted.

图 1.50 磁盘拷贝命令的帮助信息

在图 1.50 中，如果用户单击 Options 菜单，会看到图 1.51 的选项。


My Computer
Network Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Windows Help
Help Topics
Back
Annotate...
Copy
Print Topic...
Font
Keep Help on Top
Use System Colors
Start
Windows Help
9:59 AM

To make a copy of a

1 In My Computer, click I to copy.

2 On the File menu, click

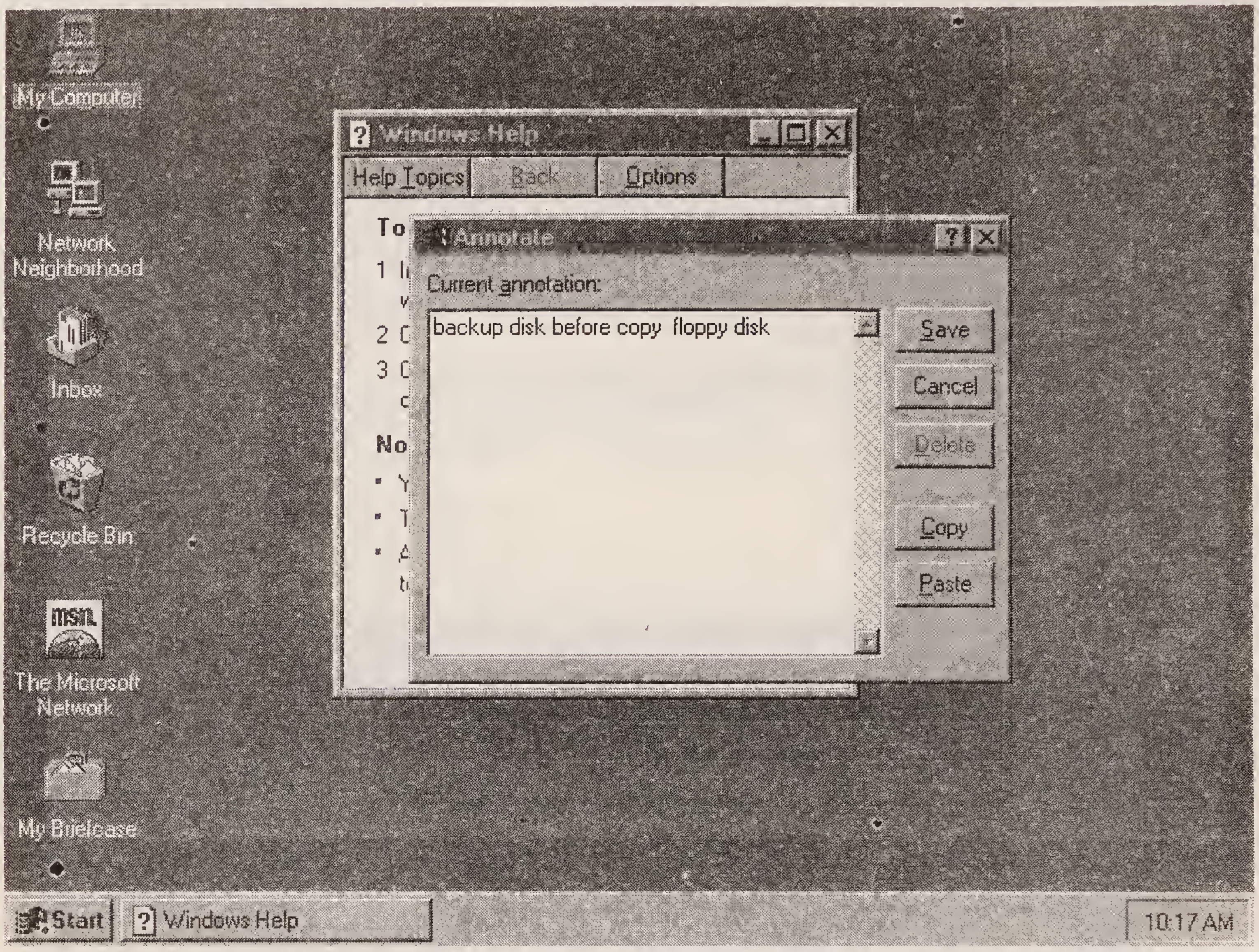
3 Click the drive you war you want to copy to, and then click Start.

Notes

- You can use the same drive for both disks.
- The disks must be the same type.
- Any existing information on the disk you copy to will be deleted.

图 1.51 Options 窗口

1）Annotate：向帮助主题添加注释以备将来使用，或为别人参考。单击它，出现一个添加帮助注释窗口，用户可以键入有关的注释，单击 Save 按钮保存注释，单击 Delete 删除注释，见图 1.52。


My Computer
Network Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Windows Help
Help Topics
Back
Options
Annotate
Current annotation:
Save
Cancel
Delete
Copy
Paste
Start
Windows Help
10:17 AM

backup disk before copy floppy disk

图 1.52　添加帮助注释窗口

2）Copy：在剪贴板上复制文本。

3）Print Topic：打印当前帮助信息。

4）Font：改变字体大小。

5）Keep Help on Top：实时帮助。

6）Use System Colors：使用系统颜色。

（3）查找帮助形式

单击 Find，显示出查找帮助窗口，见图 1.53。在第一个区域内键入用户想查找有关帮助的关键词，在第一个列表框内选择与域内相匹配的单词可缩小查找范围。在第二个列表框中双击有关选项就可获取具体的帮助信息。例如：在第一个区域中键入“Cop”，再在第一个列表框中单击与之相匹配的 Copy 项。在第二个列表框中显示与 Copy 相关帮助主题，双击 Copying a file or folder（见图 1.54）就获取拷贝文件和文件夹的信息，见图 1.55。

1.6.2　其他的帮助形式

在上节介绍的是获取 Windows 95 一般帮助信息的方法，本小节介绍下列几种获取特定项目的帮助信息的方法：

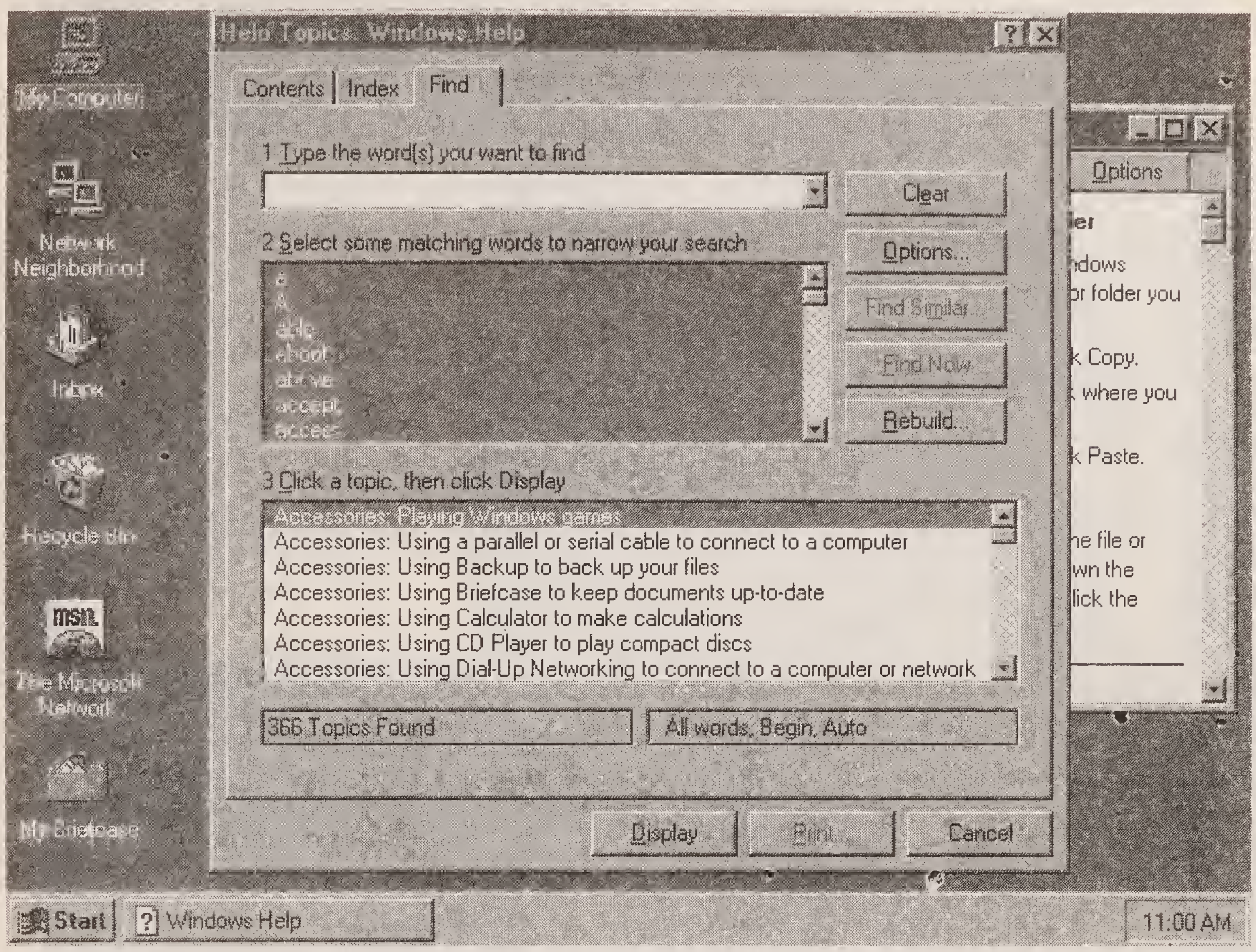

Help Topics: Windows Help
Contents
Index
Find
1 Type the word(s) you want to find
Clear
2 Select some matching words to narrow your search
Options...
Find Similar...
Find Now
Rebuild...
a
A
able
about
above
accept
access
3 Click a topic, then click Display
Accessories: Playing Windows games
Accessories: Using a parallel or serial cable to connect to a computer
Accessories: Using Backup to back up your files
Accessories: Using Briefcase to keep documents up-to-date
Accessories: Using Calculator to make calculations
Accessories: Using CD Player to play compact discs
Accessories: Using Dial-Up Networking to connect to a computer or network
366 Topics Found
All words, Begin, Auto
Display
Print...
Cancel
My Computer
Network Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Options
Start
Windows Help
11:00 AM

图 1.53　查找帮助窗口

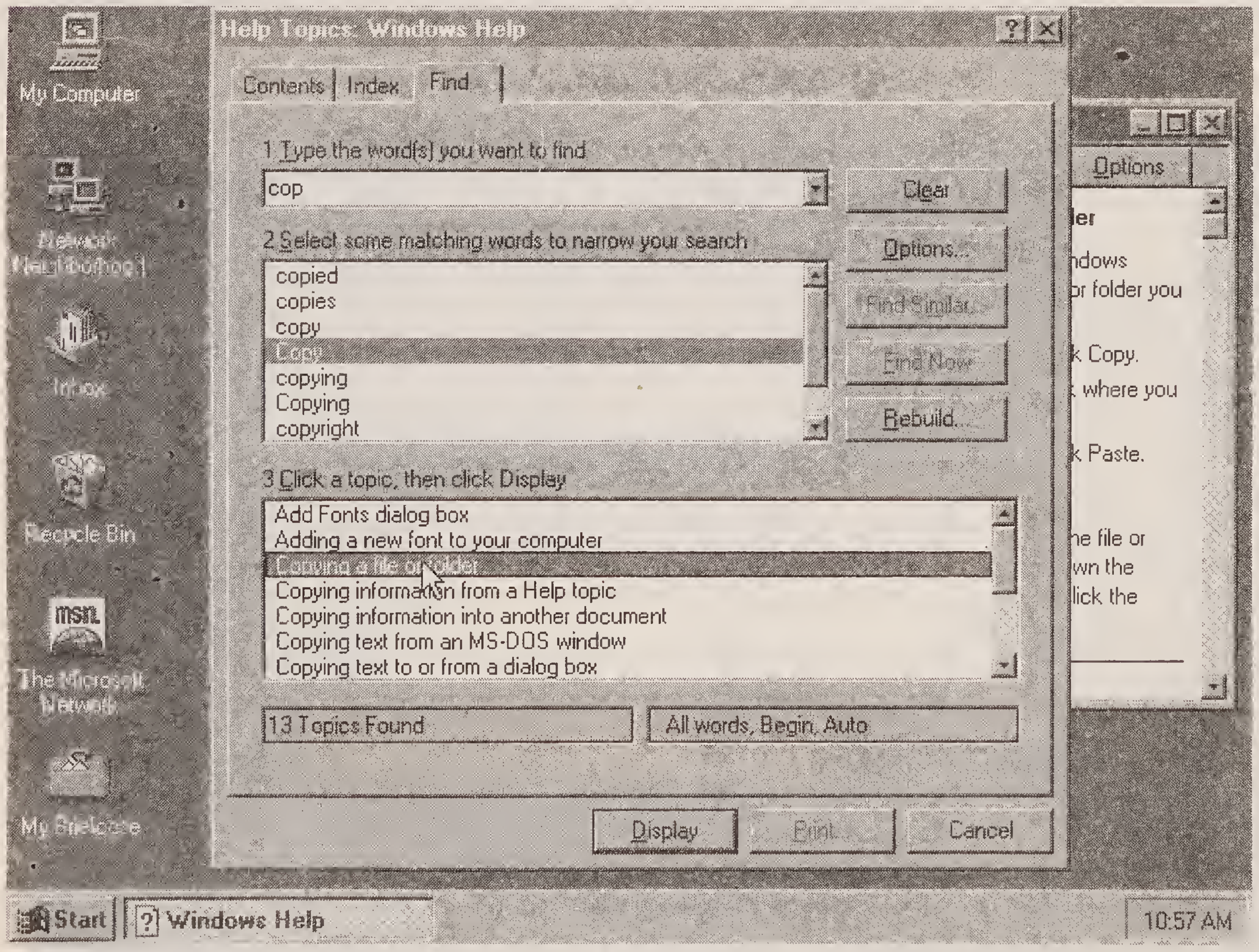
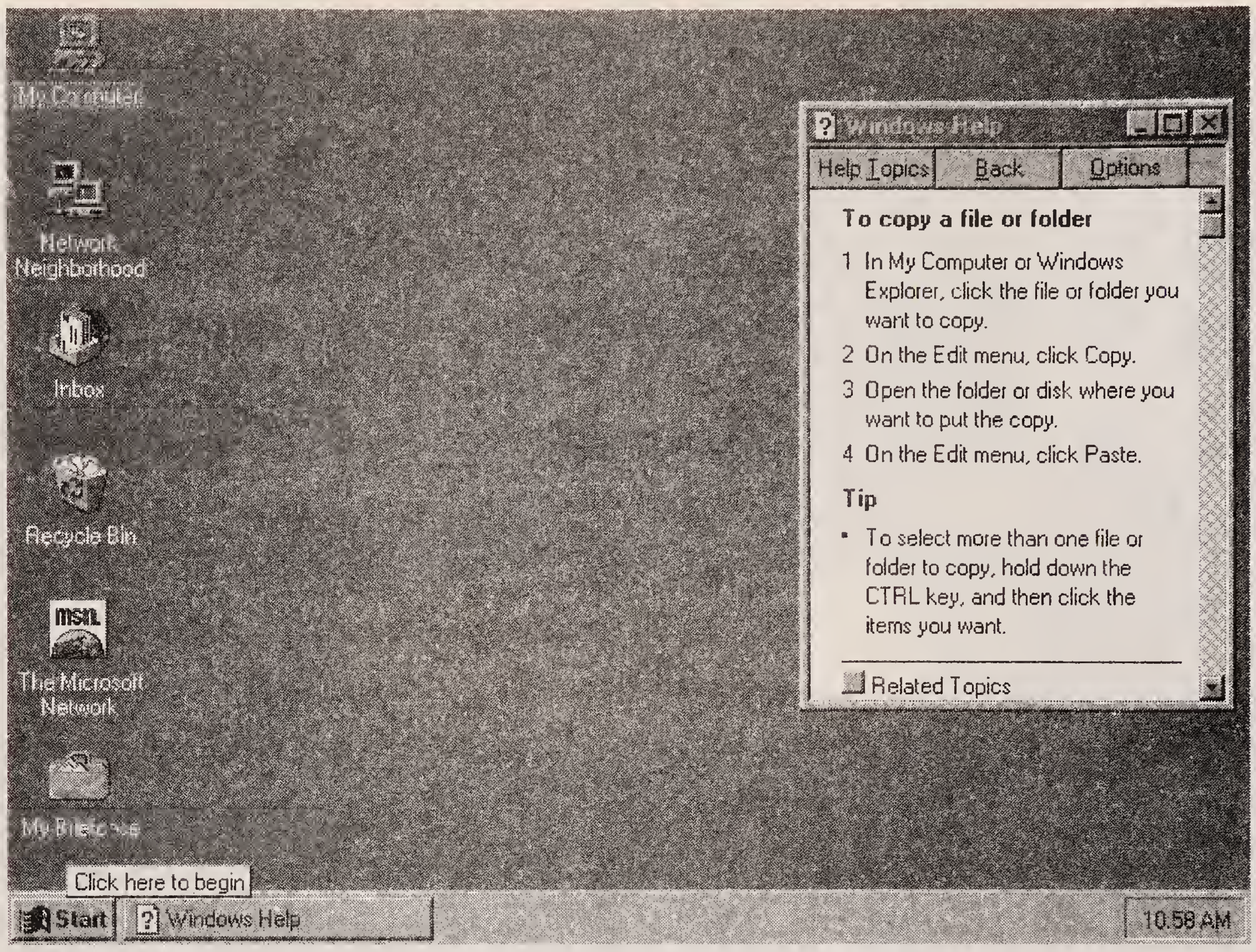

Help Topics: Windows Help
Contents
Index
Find
1 Type the word(s) you want to find
cop
Clear
2 Select some matching words to narrow your search
Options...
Find Similar...
Find Now
Rebuild...
copied
copies
copy
Copy
copying
Copying
copyright
3 Click a topic, then click Display
Add Fonts dialog box
Adding a new font to your computer
Copying a file or folder
Copying information from a Help topic
Copying information into another document
Copying text from an MS-DOS window
Copying text to or from a dialog box
13 Topics Found
All words, Begin, Auto
Display
Print...
Cancel
My Computer
Network Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Options
Start
Windows Help
10:57 AM

图 1.54　键入关键词


My Computer
Network Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Windows Help
Help Topics
Back
Options
Related Topics
Click here to begin
Start
Windows Help
10:58 AM

To copy a file or folder

1 In My Computer or Windows Explorer, click the file or folder you want to copy.

2 On the Edit menu, click Copy.

3 Open the folder or disk where you want to put the copy.

4 On the Edit menu, click Paste.

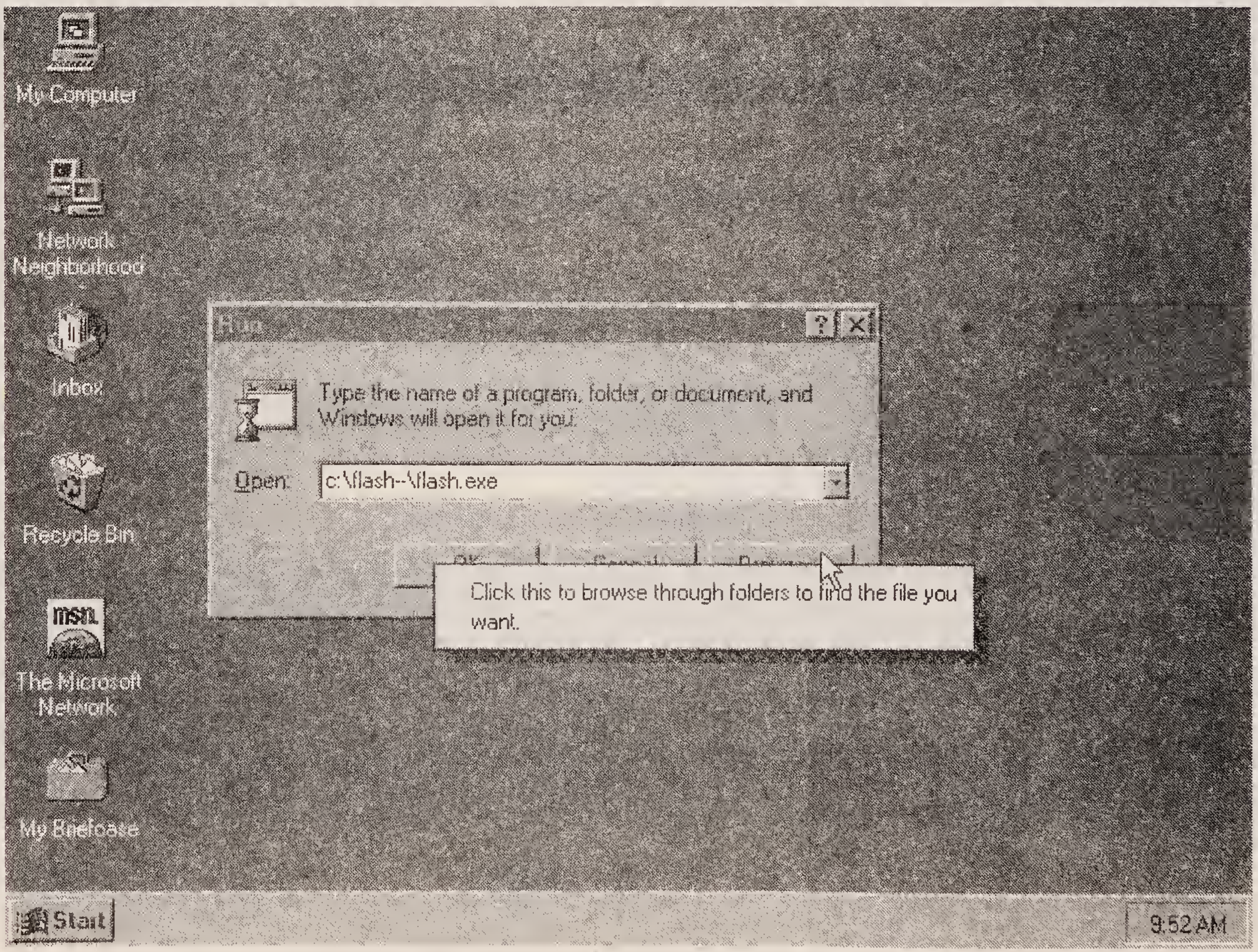
Tip

- To select more than one file or folder to copy, hold down the CTRL key, and then click the items you want.

图 1.55　拷贝文件和文件夹的帮助信息

1）同 Windows 95 前期版本一样，可以通过 F1 键获取某项的帮助信息。

2）在有的应用程序中，在标题栏中有一个“?”图标。单击此图标，鼠标指针改变成一个问号标记，用户可以在对话框上的任何项上放置，单击后即可获取该项的帮助信息，见图 1.56。


My Computer
Network Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Run
OK
Cancel
Browse...
Start
9:52 AM

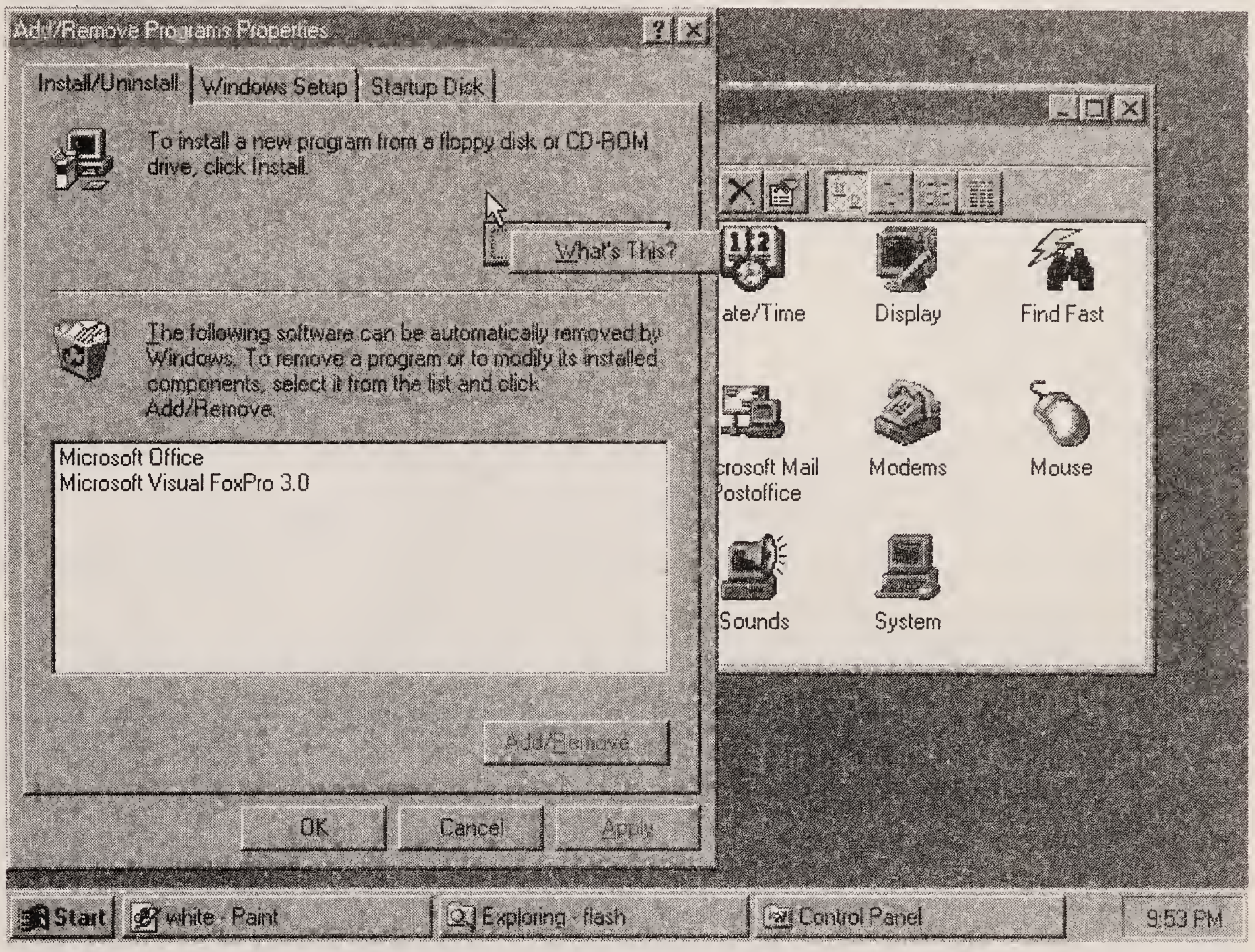
Type the name of a program, folder, or document, and Windows will open it for you.

Open: c:\flash--\flash.exe

Click this to browse through folders to find the file you want.

图 1.56　利用“?”图标获取特定项的帮助信息

3）使用鼠标右按钮单击屏幕上某项之后显示“What is this”，单击“What is this?”即可获取帮助信息，见图 1.57 和图 1.58。



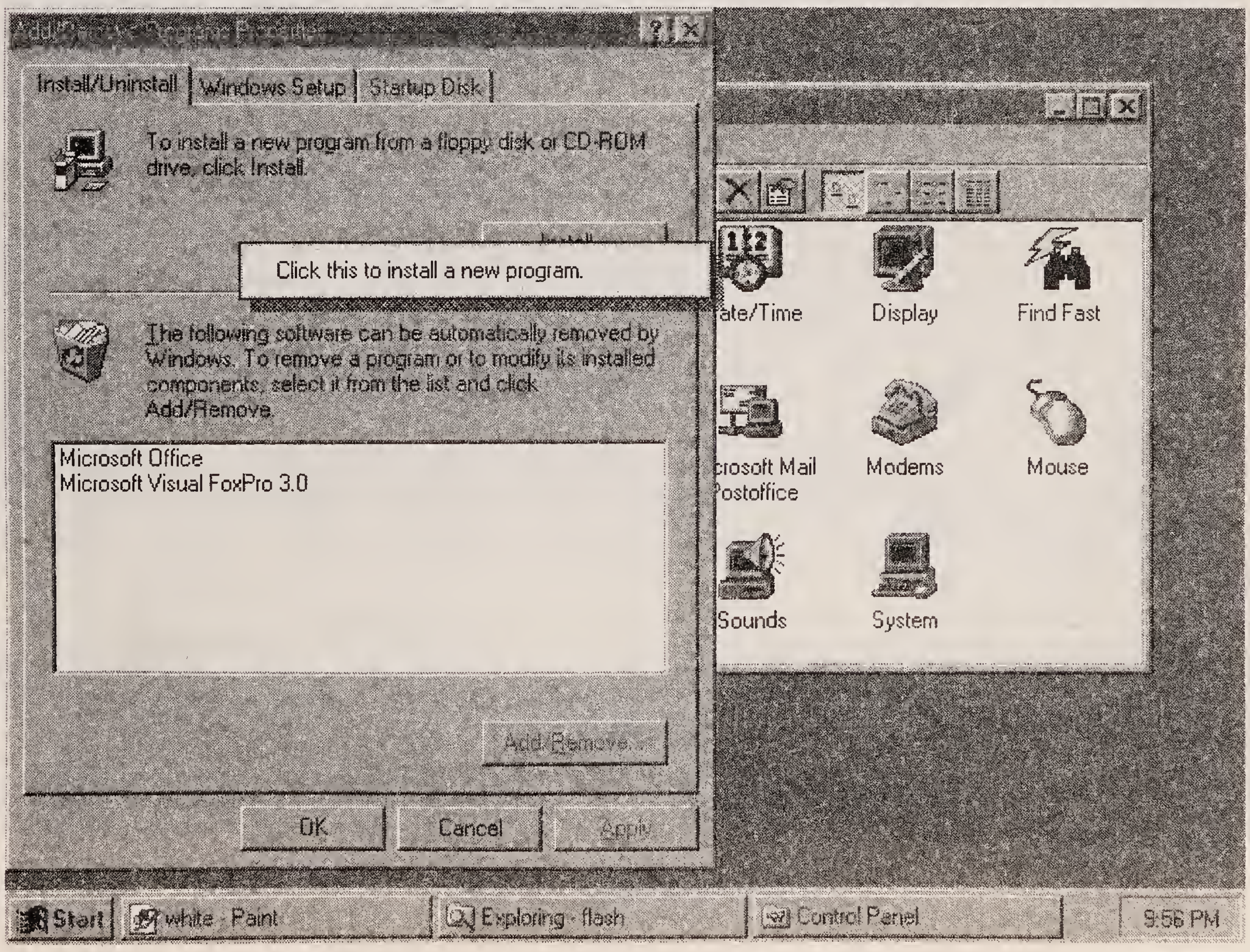
To install a new program from a floppy disk or CD-ROM drive, click Install.

The following software can be automatically removed by Windows. To remove a program or to modify its installed components, select it from the list and click Add/Remove.

Microsoft Office
Microsoft Visual FoxPro 3.0

Add/Remove Programs Properties
Install/Uninstall
Windows Setup
Startup Disk
What's This?
Add/Remove
OK
Cancel
Apply
ate/Time
Display
Find Fast
crosoft Mail Postoffice
Modems
Mouse
Sounds
System
Start
white - Paint
Exploring - flash
Control Panel
9:53 PM

图 1.57 显示“What's this?”



To install a new program from a floppy disk or CD-ROM drive, click Install.

Click this to install a new program.

The following software can be automatically removed by Windows. To remove a program or to modify its installed components, select it from the list and click Add/Remove.

Microsoft Office
Microsoft Visual FoxPro 3.0

Install/Uninstall
Windows Setup
Startup Disk
Add/Remove...
OK
Cancel
Apply
ate/Time
Display
Find Fast
crosoft Mail Postoffice
Modems
Mouse
Sounds
System
Start
white - Paint
Exploring - flash
Control Panel
9:56 PM

图 1.58 显示帮助信息

4）工具提示：工具提示是指在鼠标指向任务栏或应用程序的工具条上就显示有关项的联机帮助信息。

1.6.3 查找功能

在使用 Windows 95 的过程中，有时候需要知道某个文档或文件夹在何处，用户可以用 Start 菜单上的 Find 命令执行查找功能。Windows 95 提供的查找功能是一个帮助定位文件、文件夹和计算机的查找工具，它可以搜索本系统上的以及网络上的驱动器，也可以搜索网络中的某台计算机。

用户单击 Start 菜单，把鼠标指针指向 Find，弹出一个菜单，如图 1.59 所示，系统显示选项，而且选中任何一项都能打开一个对话框。Windows 95 的 Find 相当于 Windows 3.x File Manager 中的 File search，但性能有了很大提高。Find 可以查找文件或文件夹；可以查找文件或文件夹名的一部分；可以根据上次修改数据查找文档；可以查找文档中的文本；保存查找结构；还可以处理查找到的文件，例如打开、打印和快速查看。

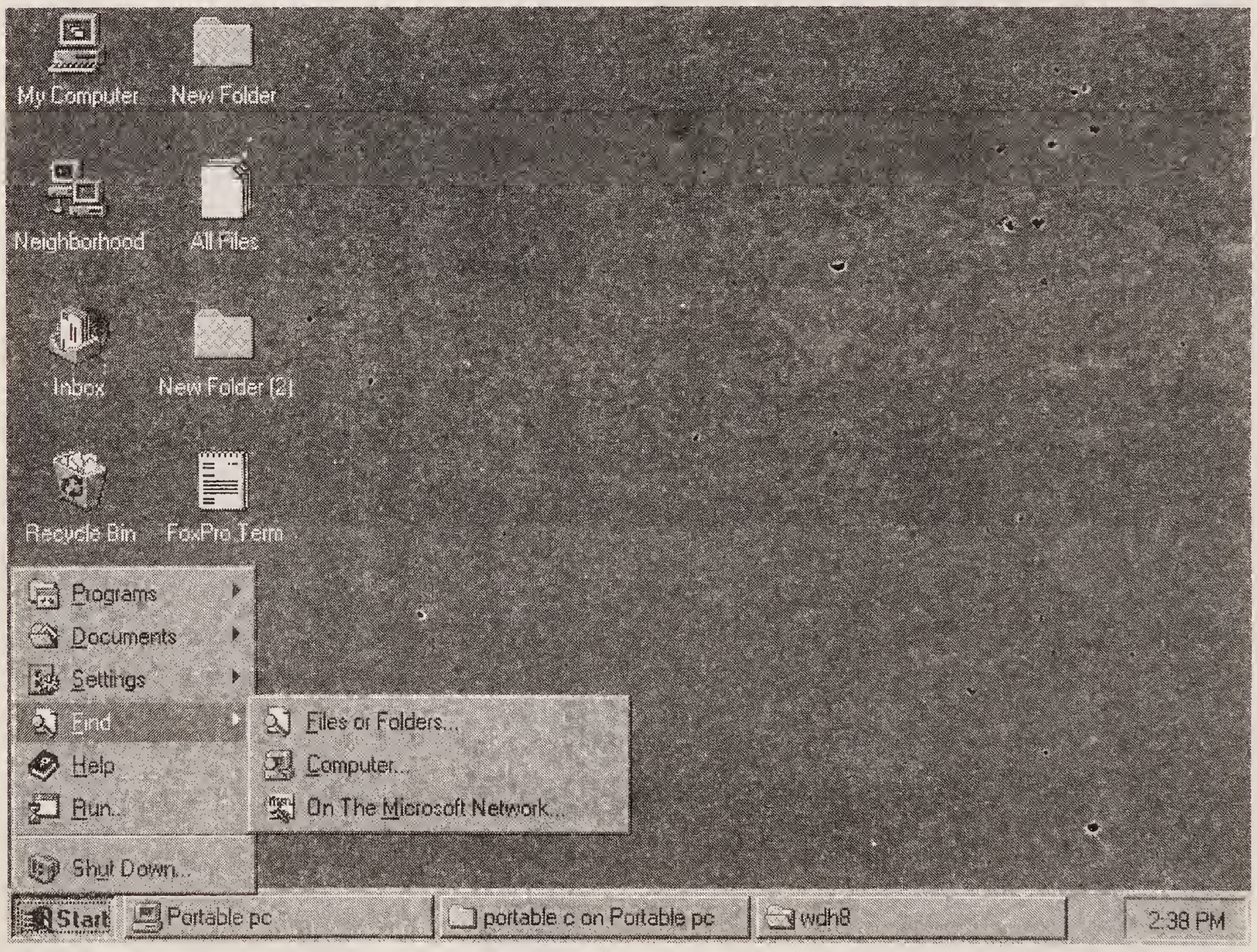
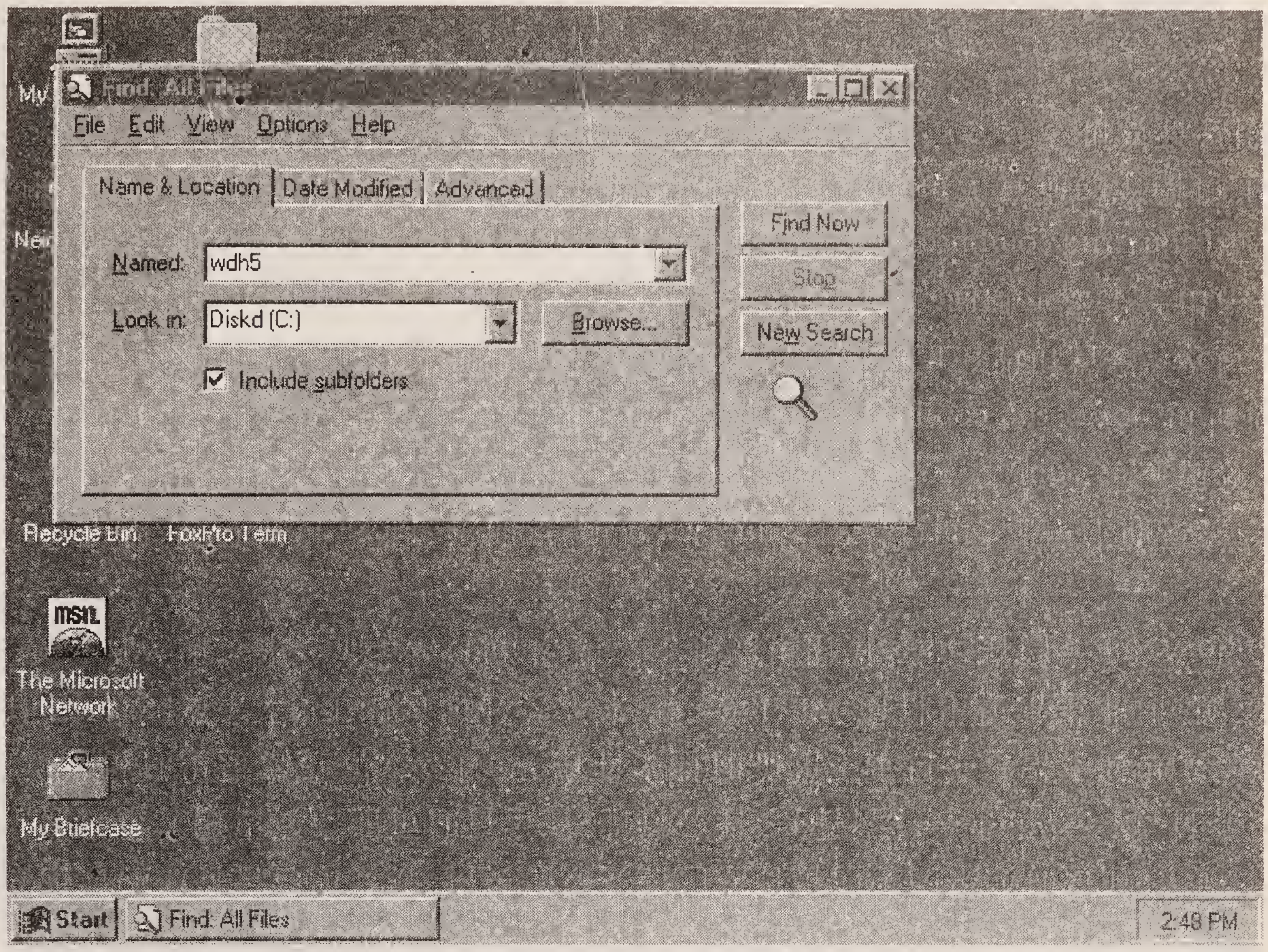

My Computer
New Folder
Neighborhood
All Files
Inbox
New Folder (2)
Recycle Bin
FoxPro Term
Programs
Documents
Settings
Find
Help
Run...
Shut Down...
Files or Folders...
Computer...
On The Microsoft Network...
Start
Portable pc
portable c on Portable pc
wdh8
2:38 PM

图 1.59 鼠标指针指向 Find 的系统显示

用户单击“Files or Folders”，屏幕上弹出查找窗口，如图 1.60 所示。在 Named 框内键入欲查找的文件或文件夹名，再在 Lookin 框内指定查找路径。

单击 Find Now 按钮执行查找功能，图 1.61 为查找到的结果。用户单击 New search 又可以开始新的查找。



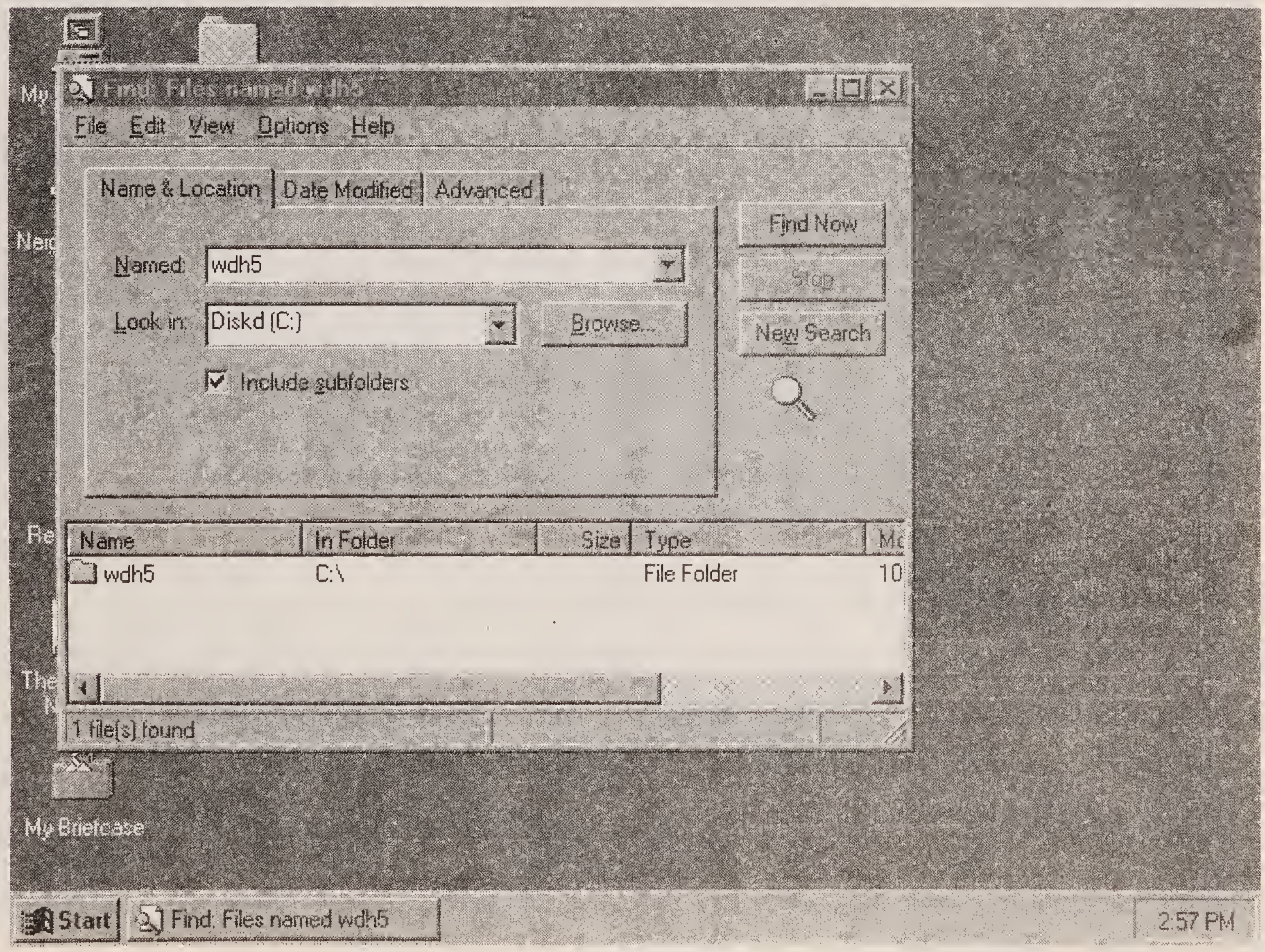
Find: All Files
File　Edit　View　Options　Help
Name & Location　Date Modified　Advanced
Find Now
Stop
New Search
Browse...
Recycle Bin
The Microsoft Network
My Briefcase
Start
Find: All Files
2:48 PM

Named: wdh5

Look in: Diskd (C:)

☑ Include subfolders

图 1.60　Find 窗口



Find: Files named wdh5
File　Edit　View　Options　Help
Name & Location　Date Modified　Advanced
Find Now
Stop
New Search
Browse...
My Briefcase
Start
Find: Files named wdh5
2:57 PM

Named: wdh5

Look in: Diskd (C:)

☑ Include subfolders

Name	In Folder	Size	Type	Mc
wdh5	C:\		File Folder	10

1 file(s) found

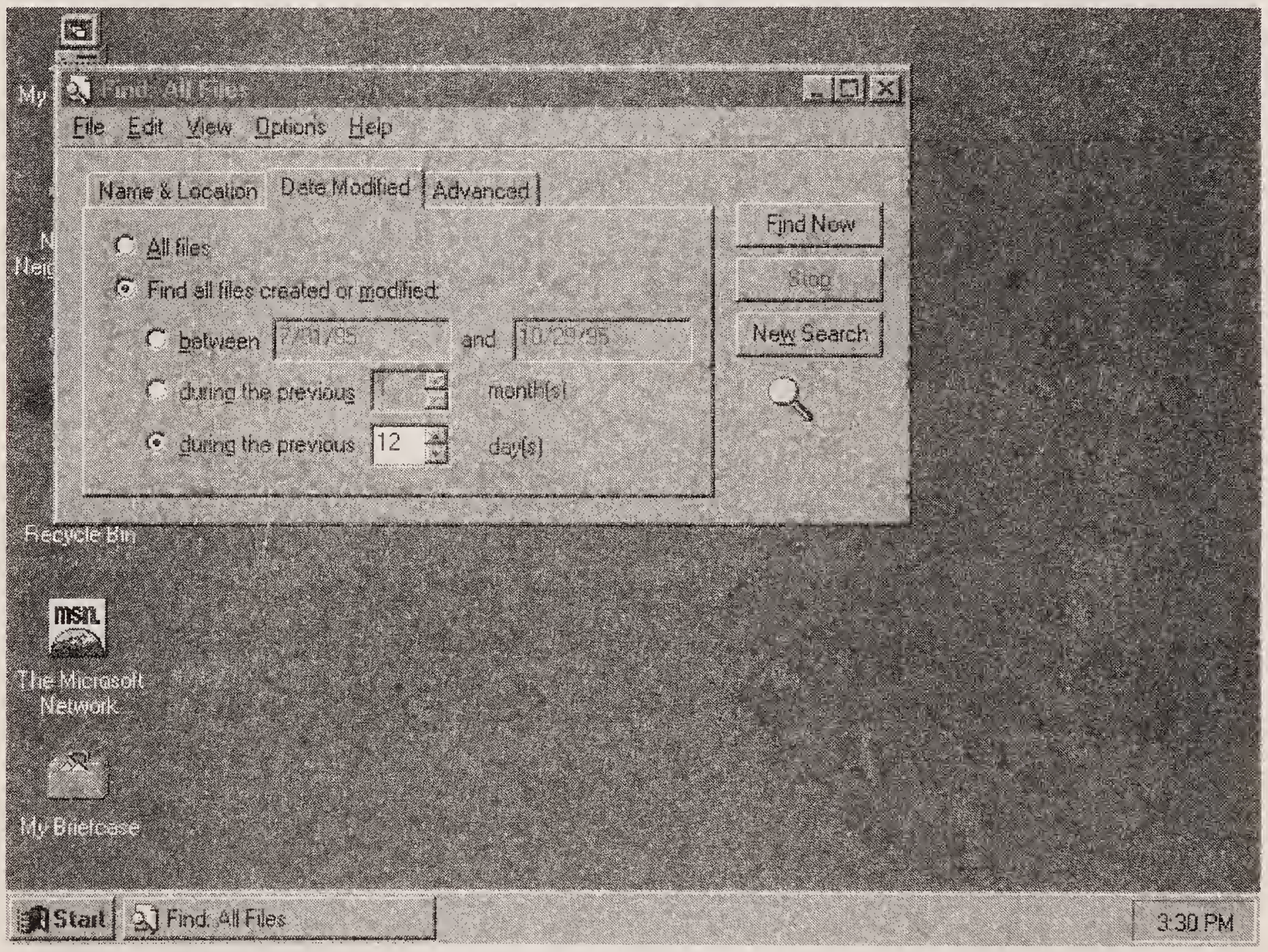
图 1.61　查找结果的显示

如果用户忘记要查找的路径，可以用 Browse 按钮浏览有关路径。如果用户不键入文件名则默认查找所有文件或文件夹。如果用户只知道文件或文件夹一部分，也可以进行查找。例如，键入文件名 Windows 95，查找结果将显示所有包含在 Windows 95 的文件或文件夹名。如果用户键入 *.EXE，则将在指定路径中查找所有带 EXE 扩展名的文件。

除了用上述方法查找外，还可以用“Find”窗口的“Data Modified”和“Advanced”标签来查找，下面分别介绍。

(1) 利用“Data Modified”标签查找文件或文件夹

选择“Data Modified”标签，按照屏幕提示进行操作，见图 1.62。例如选中“Find all files created or modified”单选按钮，再选中“during the previous (days)”单选按钮，键入用户需要查找的天数，最后单击 Find Now 执行查找功能，图 1.63 显示某一查找结果。

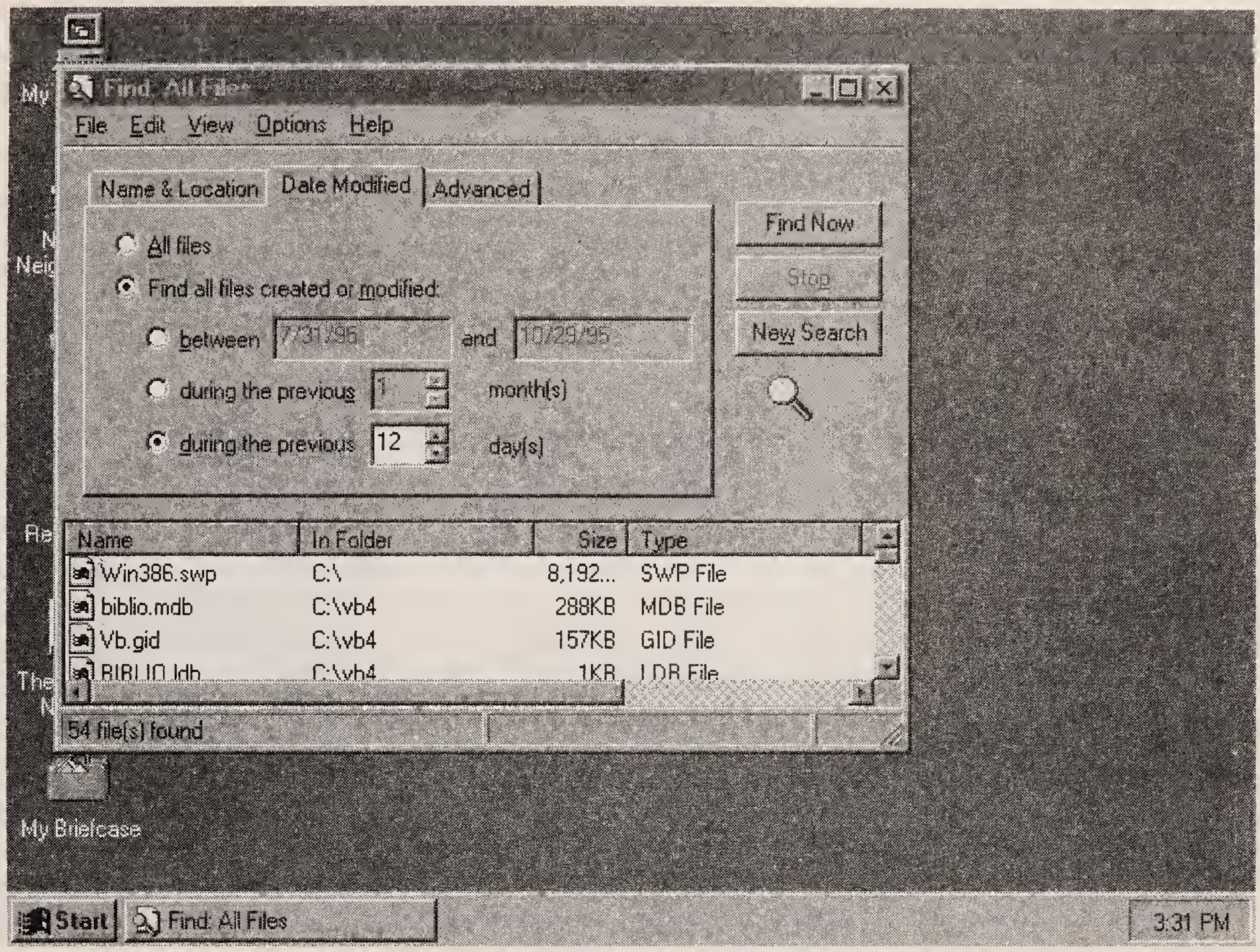

○ All files
● Find all files created or modified:
○ between 7/01/95 and 10/29/95
○ during the previous 1 month(s)
● during the previous 12 day(s)

Find: All Files
File　Edit　View　Options　Help
Name & Location
Date Modified
Advanced
Find Now
Stop
New Search
Recycle Bin
The Microsoft Network
My Briefcase
Start
Find: All Files
3:30 PM

图 1.62　利用“Data Modified”标签查找文件或文件夹

(2) 利用“Advanced”标签查找文件或文件夹

选择“Advanced”标签，显示“Advanced”标签的 Find 窗口，见图 1.64。在 File Type 下拉式列表框中选择文件所属的类型，接下来的“Containing text”文本框中键入待查找的文件或文件夹中包含的文字（当查找该类型所有文件或文件夹时不用此选项）。在下面的“sizes”文本框只有两种选择：at least 和 at most。at least 的含义是：如果在后面的 KB 框内选择输入 100K，那么将查找所有等于 100KB 或大于 100KB 的文件或文件夹。at most 的含义刚好相反。图 1.65 是某一查找结果。

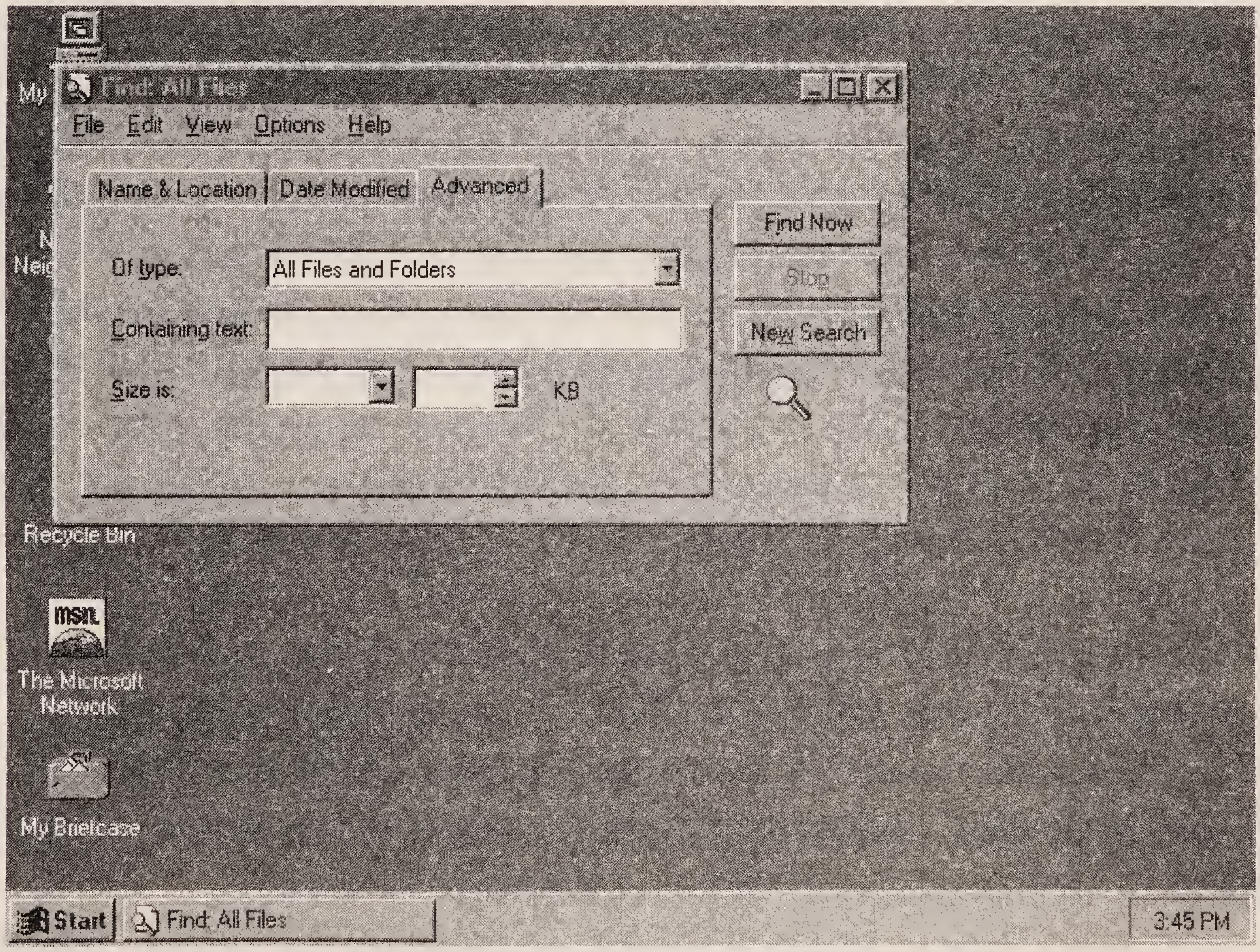

Find: All Files
File
Edit
View
Options
Help
Name & Location
Date Modified
Advanced
Find Now
Stop
New Search
My Briefcase
Start
Find: All Files
3:31 PM

○ All files
● Find all files created or modified:
○ between 7/31/95 and 10/29/95
○ during the previous 1 month(s)
● during the previous 12 day(s)

Name	In Folder	Size	Type
Win386.swp	C:\	8,192...	SWP File
biblio.mdb	C:\vb4	288KB	MDB File
Vb.gid	C:\vb4	157KB	GID File
BIBLIO.ldb	C:\vb4	1KB	LDB File

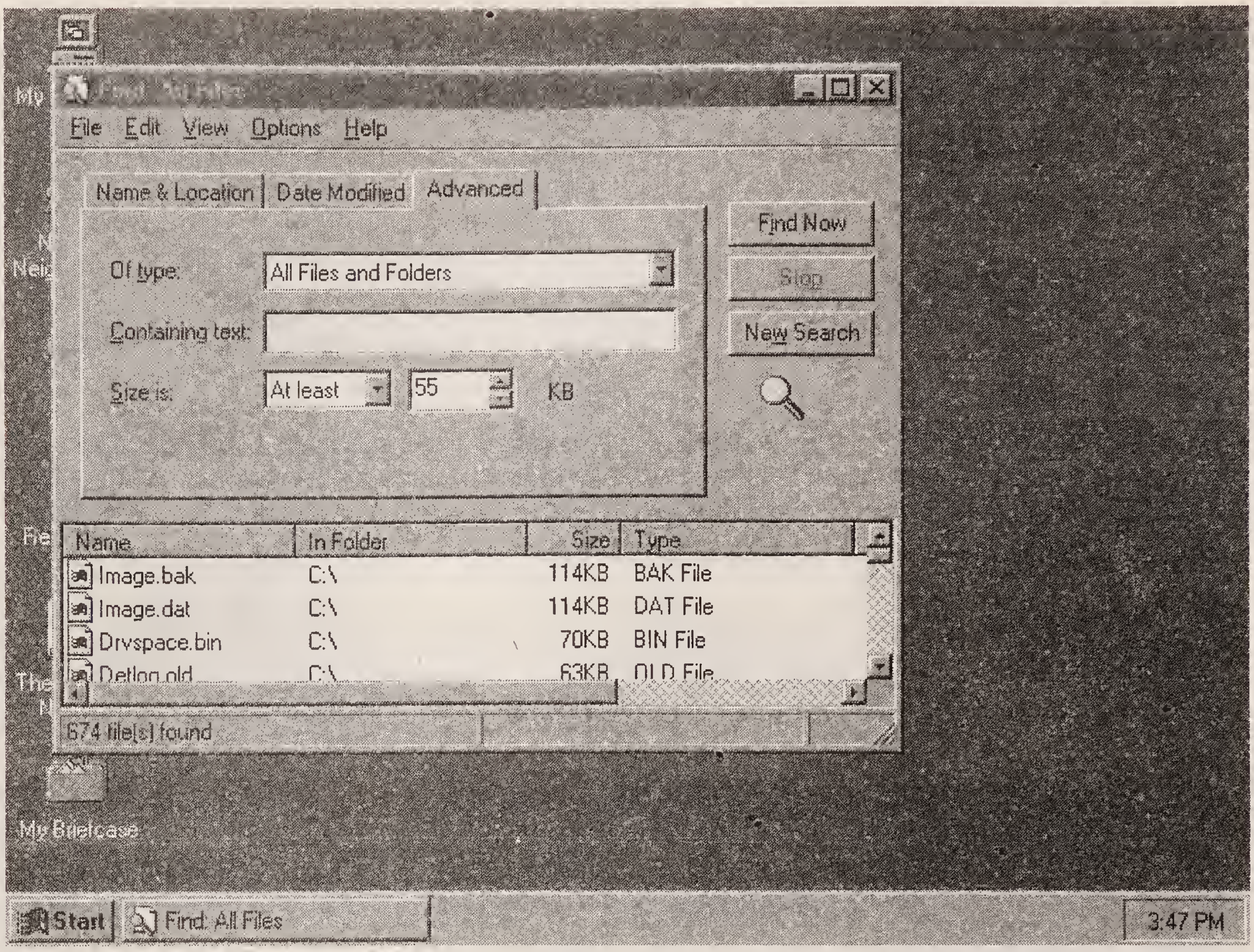
54 file(s) found

图 1.63　某一查找结果显示


Find: All Files
File
Edit
View
Options
Help
Name & Location
Date Modified
Advanced
Find Now
Stop
New Search
Recycle Bin
The Microsoft Network
My Briefcase
Start
Find: All Files
3:45 PM

Of type: All Files and Folders
Containing text:
Size is: KB

图 1.64　“Advanced”标签的查找窗口


Find: All Files
File Edit View Options Help
Name & Location
Date Modified
Advanced
Find Now
Stop
New Search
My Briefcase
Start
Find: All Files
3:47 PM

Of type: All Files and Folders

Containing text:

Size is: At least 55 KB

Name	In Folder	Size	Type
Image.bak	C:\	114KB	BAK File
Image.dat	C:\	114KB	DAT File
Drvspace.bin	C:\	70KB	BIN File
Detlog.old	C:\	63KB	OLD File

674 file(s) found

图 1.65　某一查找结果的显示

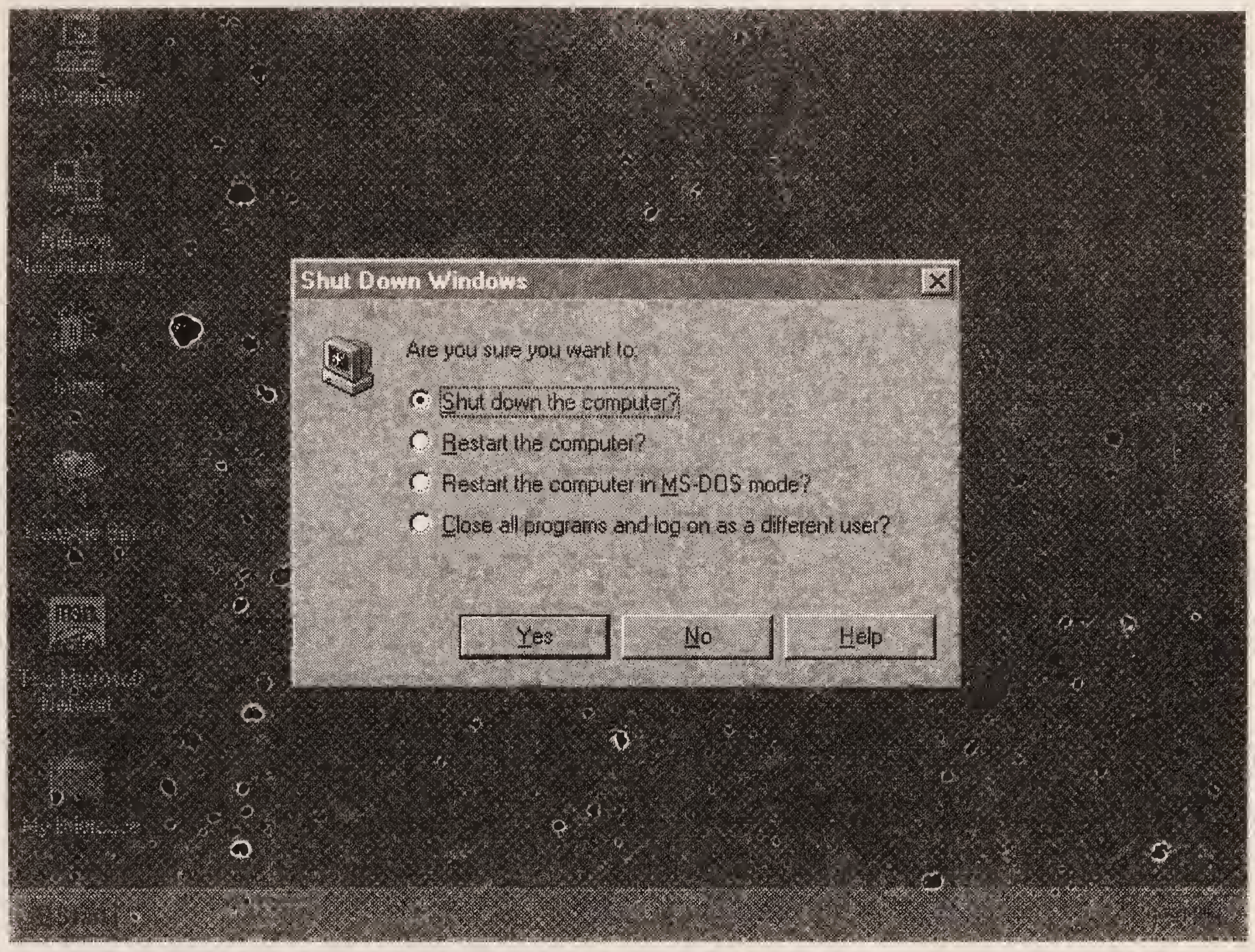
1.7　退出 Windows 95 操作系统

退出 Windows 95 操作系统不同于退出 DOS 操作系统那么简单——只要关闭电源就可以。由于 Windows 95 是一个多任务多线程的操作系统，有时同时前台运行某一程序，后台运行几个程序，在这时如果因为前台程序已经完成而关机，就会把后台程序的数据和运行结果丢失。并且，由于 Windows 运行的多任务特性，在运行时可能需要占用大量磁盘空间临时保存信息，这些处于指定子目录下的临时性文件在 Windows 95 正常退出时将予以删除，以免浪费资源，但非正常退出将使 Windows 95 来不及处理这些工作，导致硬盘空间的浪费。因此 Windows 95 专门在 Start 按钮里安排了一项 Shut Down 命令来实现 Windows 95 正常退出。

在系统桌面上单击 Start，选中 Shut Down 项，这时屏幕低亮度，以表示响应了系统的关闭命令，然后显示 Shut Down 关闭对话框，见图 1.66。对话框中给出了四个选项：

(1) Shut down the computer：选择该项用来退出系统，并准备关机，同时把本次关机的有关 Windows 设置和现在正保留在内存的修改保存到硬盘中。在 Windows 95 作好这些工作后，便给出一个提示画面，提示用户现在可以安全关闭你的计算机了。

(2) Restart the computer：该选项用于重新启动计算机，用户选择该项后，系统将保留用户本次上机操作的所有 Windows 设置，并把当前存储在内存中的信息写入硬盘中，然后自动重新引导计算机。



Shut Down Windows

Are you sure you want to:

● Shut down the computer?

○ Restart the computer?

○ Restart the computer in MS-DOS mode?

○ Close all programs and log on as a different user?

图 1.66 **Shut Down 关闭对话框**

(3) Restart the computer in MS-DOS mode：以 MS-DOS 方式重新启动计算机。用户如果想退出 Windows 95 操作系统，进入 MS-DOS 系统，可以选择该项。这样，在 Windows 95 保存所有信息后，便把控制权移交给存在于系统上的 DOS 操作系统。如果用户用完 DOS 以后，想再进入 Windows 系统。只要在 DOS 提示符下键入 Win 或 Exit，这时系统会询问用户是否要返回 Windows 系统，回答“Y”后系统便重新启动，进入 Windows 操作系统。

(4) Close all programs and log on as a different user：该选项将关闭用户的所有应用程序，并把用户的计算机退出计算机网络，从而准备系统让另一个用户使用。这适用于挂在网络上的公共计算机。

第二章　Visual Basic 4.0 简介

2.1　Visual Basic 发展简史

Basic 语言是在 1964 年由美国 Dartmouth 大学的 John G · kemeny 和 Thomas E · kurtz 两位教授共同创立的一种通用的计算机算法语言。Basic 是 Beginners All-purpose Symbolic Instruction Code（初学者通用符号指令代码）的缩写。由于其简单易学以及人机对话方便，曾得到过广泛的应用。但随着科技的不断发展，Basic 语言由于其运行速度慢等原因而逐渐被其他高级语言所代替。

随着图形用户接口（GUI）的出现，在微机工业引发了一场革命。Microsoft（微软）公司于 1988 年推出的 Windows 操作系统就由于其漂亮的图形界面和快捷方便的鼠标操作等众多优点而赢得了广大用户的青睐。但在 Visual Basic 前，开发一个 Windows 应用程序则非常困难，因为程序员必须考虑许多事情，如鼠标在做什么，怎么编写菜单等，这得需要专门的 C 程序员编写大量的程序。于是 Microsoft 公司由 1991 年推出了 Visual Basic 1.0 版本，这套软件为开发 Windows 应用程序提供了强有力的工具。

Visual Basic 在一开始就包含了 GUI 界面，它使用户可以直接设计屏幕，逐个添加用户界面，并且用户所使用的控件如命令按钮等，可以自动跟踪用户的操作。当窗口或控件发生一个事件（Event）时，Visual Basic 就让编程者自已来控制程序，即编写代码来响应这些事件。例如当我们单击鼠标来按动一个按钮时，我们并不需要知道这一过程的代码是什么以及这一过程是怎样发生的，只需要编写一段代码来告诉计算机在按钮按动以后执行什么命令就可以了。所以 Visual Basic 比以往的各种工具都简便、迅速，受到了广大应用程序设计程序员的欢迎。

当然 Visual Basic 在刚推出时也是有缺陷的，功能也相对的少一些。经过 Microsoft 公司的不断努力，又相继推出了 Visual Basic 2.0 以及 Visual Basic 3.0 版本。

随着 Windows 95 的推出，Visual Basic 的最新版本 Visual Basic 4.0 也出台了，下面我们就介绍一下 Visual Basic 4.0 的一些特点。

2.2　Visual Basic 4.0 中新增加的内容

Visual Basic 4.0 是 Visual Basic 3.0 的升级版本，是在 Visual Basic 3.0 的基础上发展起来的。它继承了 3.0 版本的许多优点，同时又增加了一些新的功能，使其运用起来更加得心应手。下面我们就来介绍一下 Visual Basic 4.0 增加的一些新功能：

（1）Visual Basic 版本

Visual Basic for Application 版本在 2.0 版本的基础上增加了一个语言引擎（language engine）。4.0 版本与早期的 Visual Basic 版本是向后兼容的，与 Microsoft Excel 5.0 版本以及

Microsoft Project 4.0 版本中的 Visual Basic for Application 版本也是兼容的。而 Visual Basic for Application 版本 4.0 则使得进行 OLE（Object Linking and Embedding 对象的动态链接与嵌入）变得更加容易，并且 Visual Basic 代码在支持 OLE 动态应用程序的模块之间的转移方面也更加容易。

（2）OLE 自动化

使用 Visual Basic 中的 OLE 自动化时，可以在应用程序里通过控制其他应用程序中的对象来借用它们的某些功能。如果这个对象是一个 OLE 自动化对象，就可以通过代码的方式来利用它的属性和方法。OLE 提供给某一程序与其他程序动态分享信息的能力，使用户可以在当前应用程序里收集和运用其他应用程序的数据与功能。例如，可以建立一个应用程序，在这个应用程序里，利用 Misrosoft Excel 建立一个计算器，用 Microsoft Word 建立一个报表（reports）。为了使这种操作变为更加容易方便的拖放方式，还可以直接把一些对象，例如 Microsoft Excel 电子表格作为一个控件加到工具箱（ToolBox）里。

（3）可重复使用的对象和集合

Visual Basic 4.0 允许用户建立一些可重复使用的对象，这些对象各自有自己的属性和方法，并被集中到一个对象模块中。我们把这些对象定义为类（class），对象模块被定义为新类模块（new class module）。用户还可以建立自己的集合（collection）。

（4）属性过程（property procedure）

属性过程允许用户把定制（custom）属性添加到窗体（Form）和模块中，并且当这些属性被设置或被检索时执行相关代码。例如，用户可以把一个反白（Inverted）属性添加到一个 Form 窗口中。当反白属性被设置为真（true）时，在相关属性过程中的代码就调用一个应用程序接口（API，Application Programming Interface 的缩写）在 Form 窗口中来反白一个位图（bitmap）。

（5）对象浏览器（Object Browser）

对象浏览器将从 Visual Basic 和 OLE 控件中为应用程序列出所有的类模块、属性和方法。这样，用户就可以很容易的选择对象，并把它们剪切、粘贴到代码中。

（6）增强的开发环境

在 Visual Basic 4.0 中，用户可以为自己的窗体、控件以及代码设置弹出式菜单，这种菜单只有在进行程序设计时才能进行控制，并且通过单击鼠标标右键弹出。用户还可以把 add-ins（添加与嵌入）加到源代码控件以及其它特性中去，在 Visual Basic 专业版本里，用户甚至可以创建自己的 add-ins。

（7）32 位支持以及条件编译

运用条件编译，可以把特定的平台代码段嵌入 # If... Then... # Else 语句中，并且可以把应用程序设置为 16 位或 32 位的。32 位的 Visual Basic 版本在所有工程文件中均支持长文件名，并且突破了对属性以及 Visual Basic 语言的 64K 容量限制。

（8）新的数据约束（data-bound）控件

用户能够把数据约束给更多的控件，包括在 Visual Basic 4.0 中新添的数据约束列表框（DBList）、数据约束组合框（DBCombo）以及数据约束网格（DBGrid）控件。OLE 包容器控件也是一个数据约束控件。这些控件提供了自动的列表管理和对外部数据迅速的读写访问。

（9）Jet 2.5 数据库

在 Visual Basic 4.0 中，Microsoft Jet 2.5 数据库提供了许多新的特色。在包括级联式的更新和删除功能的专业版本中，用户可以利用完整性的参考进行程序设计，并且提高了查询的速度。如果用户有一个 Microsoft Access 2.0 的拷贝，就可以用 Microsoft Jet 2.5 数据库安全地创建和修改设置。

2.3　Visual Basic 用到的几个基本术语

Visual Basic 是一种面向对象的程序设计方法，这种方法将代码和数据集成到一个独立的对象中去，当运用这个对象来完成某项任务时，我们并不需要知道这个对象是怎样工作的，只需要编写一段代码，来简单地给其传递一些消息就可以了。对于这种程序设计方法，我们将会用到一些专门的技术用语，下面我们就先简单地介绍几个最基本的术语，其他的术语在以后我们用到时再给大家逐一介绍。

(1) 对象(Object)

Visual Basic 使用的对象不再是一种抽象的东西，而是一种类似于按钮、工具框之类的真实的、可触摸的东西，我们可以把它理解为具有某些特性的可用的东西。Visual Basic 生成一个对象很容易，它不像其他语言那样需要一大堆对象命令去定义代码和数据块作为对象。

Visual Basic 中的主要对象是窗体(Form)窗口，这个窗口在启动 Visual Basic 后我们会看到它。在这个窗口里我们可以生成某些应用程序，可以把某些控件，如图片框(Picture box)、文本框(Text box)、命令按钮(Command button)或标签(Label)等等加上去。Visual Basic 应用程序的整个可视界面都是画出来的，而不是用代码编出来的。

(2) 事件(Event)

我们把生成的对象所要完成的任务，即响应对象的动作称为事件。对象与对象之间，对象与系统之间以及对象与程序之间的通信都是通过事件来进行的。当鼠标指针被移动时，系统将跟踪鼠标指针的位置，当鼠标单击一下按钮时，系统将跟踪到指针所指的对象上。如果对象是一个按钮，系统就给按钮发送一个 Click 事件，并在窗体上模拟一个按钮被按下的动作，同时向程序传递一个 Button _ click 事件，如果我们编写了一个处理 Button _ click 事件的程序代码，系统则执行这一代码所描述的过程。执行结束以后，控制权又交还给系统，并等待下一个事件。

对象并不是把所有的事件都传递给程序，许多事件都是由对象或 Windows 操作系统直接处理的。例如，当我们用鼠标拖动窗口的边界来改变其大小时，是由操作系统直接处理的，尽管窗口会给程序员传递一个 Resize 事件，以通知程序员发生了一次修改操作，但实际上程序员并不需要知道进行了什么处理。

(3) 属性(Property)

对象中的数据就保存在属性中，所有 Visual Basic 程序中的对象都有属性。例如文本框中的文本为 Text 属性，命令按钮的属性为 Command 属性，文本框使用的字体为 Fontname 属性。我们可以通过控制改变控件的属性在程序中起作用。例如我们可以通过改变文本框的 Text 属性在文本框中设置文本，还可以通过改变窗体的 Caption(标题)属性来改变标题条中的标题。

我们通过属性还可以改变对象本身的一些参数，如窗口的大小设置、背景颜色设置以及窗口是否隐藏等。对于这些属性，我们将在后面几章再作介绍。

（4）方法(Method)

对象不仅仅包含代码段，还包含数据和对这些数据进行操作的代码段。每一个对象都包含操作自己数据的代码段，我们把这就称为方法。触发一个方法，只要键入对象名以及方法名即可，例如：

Form. Print"Welcome to Visual Basic 4.0"

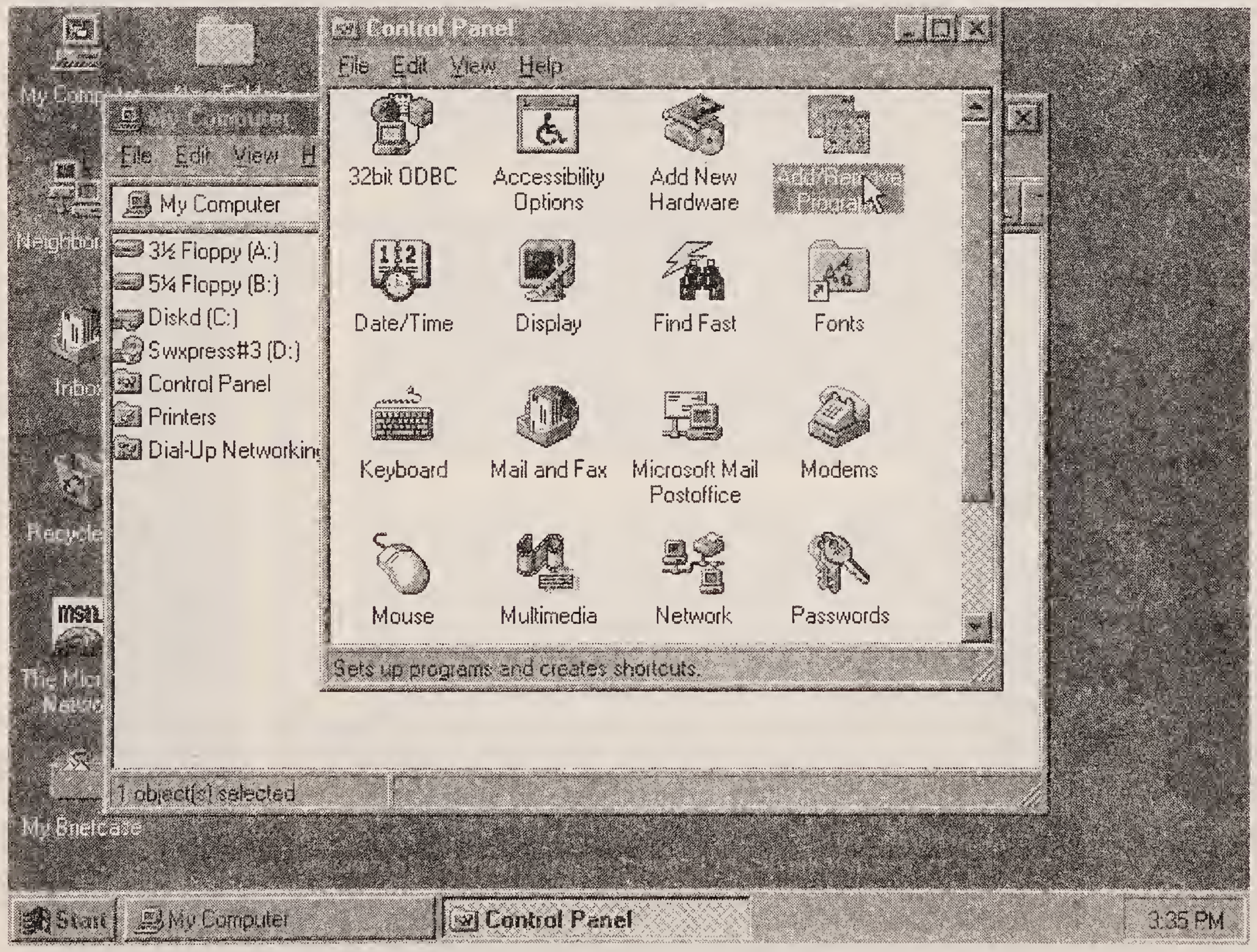
这里的 print 就是 Form 对象中包含的一个方法，它在 Form 窗口中输出文本"Welcome to Visual Basic 4.0"。

（5）过程(procedure)

过程就是一个能执行特定任务的独立的代码块。它能使程序员将一段代码组织在一起，给这段代码起一个名字，并使用这个名字来执行这段代码。在编制 Visual Basic 程序的时候，为了能将代码连接到不同的控件和格式上，必须把程序分成一个个的过程，这些事件驱动的过程将程序分成若干个任务。当发生一个事件时就执行一个相应的任务。过程不仅可以隔离与任务相关的代码，而且还可以隔离数据，即在别的过程中定义的同名的变量与本过程中的变量是互不影响的。

2.4 Visual Basic 4.0 的安装

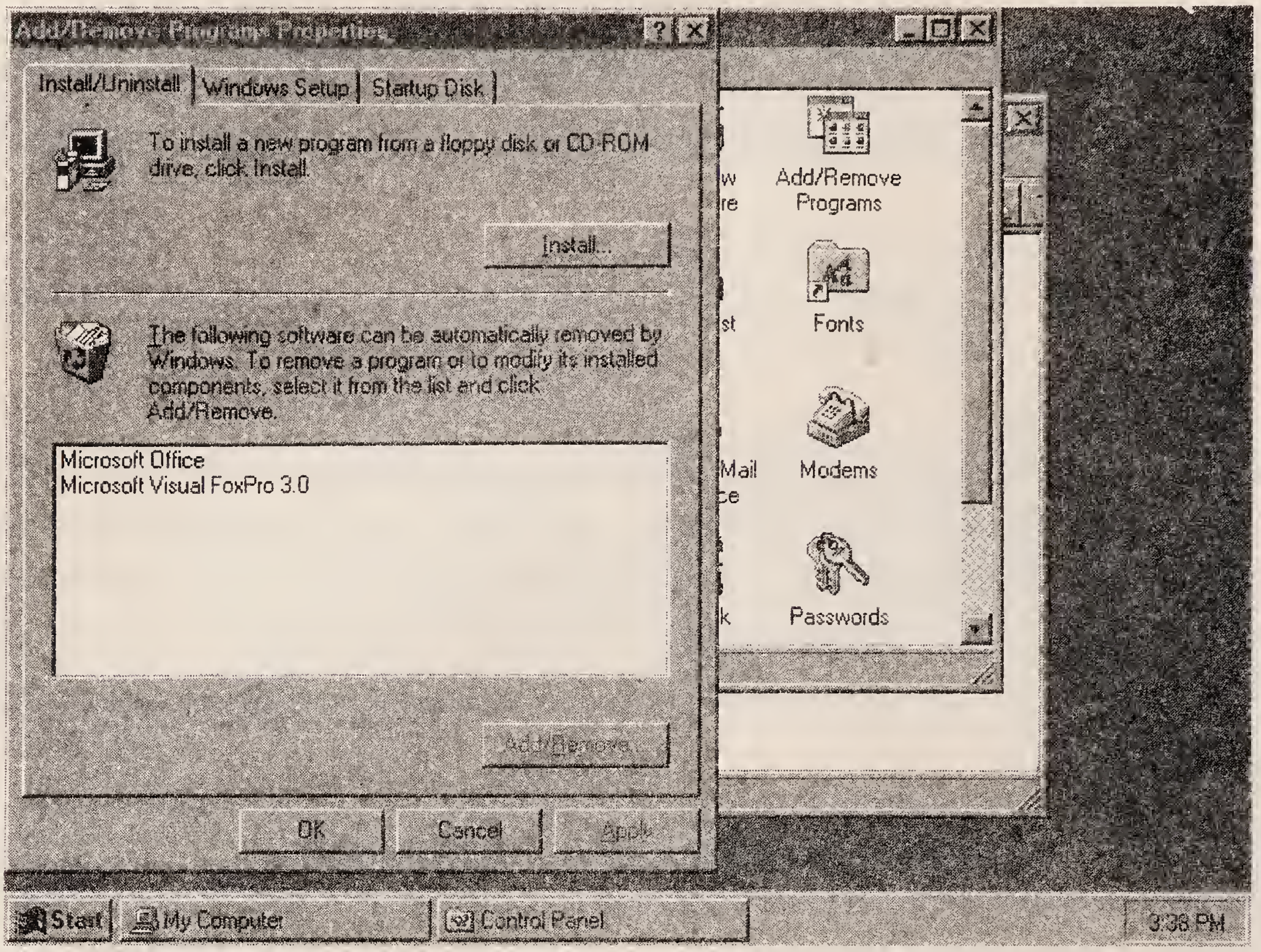
在 Windows 95 下用光盘安装 Visual Basic 4.0 即轻松又方便，用户可以用 Control Panel 上的 Add/Remove Programs 选项安装，见图 2.1。用户双击该选项，打开一对话框，见图 2.2。


Control Panel
File Edit View Help
32bit ODBC
Accessibility Options
Add New Hardware
Add/Remove Programs
Date/Time
Display
Find Fast
Fonts
Keyboard
Mail and Fax
Microsoft Mail Postoffice
Modems
Mouse
Multimedia
Network
Passwords
My Computer
File Edit View
My Computer
3½ Floppy (A:)
5¼ Floppy (B:)
Diskd (C:)
Swxpress#3 (D:)
Control Panel
Printers
Dial-Up Networking
My Briefcase
Start
My Computer
Control Panel
3:35 PM

Sets up programs and creates shortcuts.

1 object(s) selected

图 2.1 Control Panel 上安装软件选项


Add/Remove Programs Properties
Install/Uninstall
Windows Setup
Startup Disk
Install...
Add/Remove...
OK
Cancel
Apply
Add/Remove Programs
Fonts
Modems
Passwords
Start
My Computer
Control Panel
3:38 PM

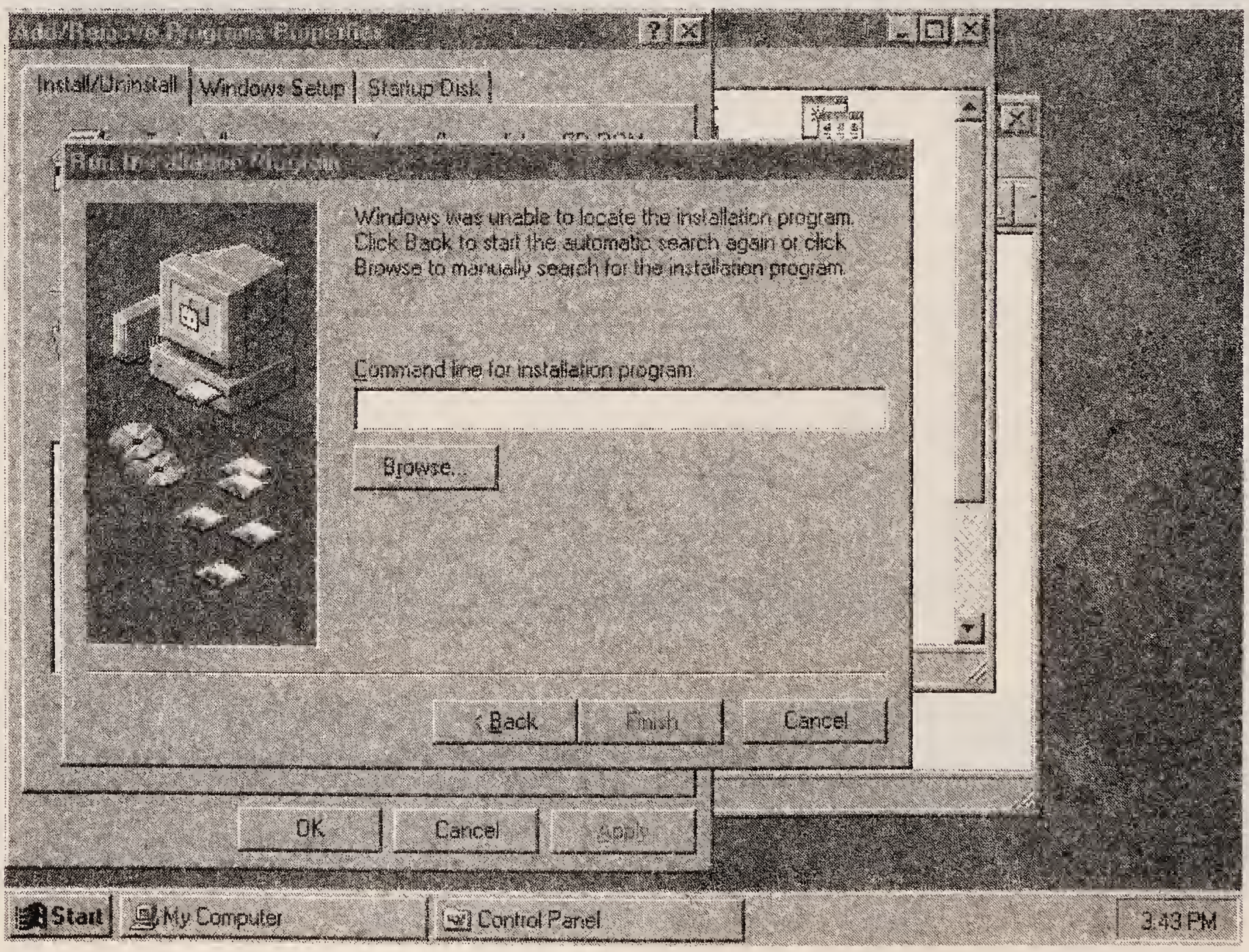
To install a new program from a floppy disk or CD-ROM drive, click Install.

The following software can be automatically removed by Windows. To remove a program or to modify its installed components, select it from the list and click Add/Remove.

Microsoft Office
Microsoft Visual FoxPro 3.0

图 2.2　Add/Remove Programs Properties 对话框

用户在图 2.2 中单击 Install 按钮，这时出现一对话框提示用户把光盘插入光盘驱动器上，然后单击 Next 按钮。在单击 Next 按钮的同时，系统寻找插有安装盘的驱动器。稍后屏幕上弹出一 Run Install Program（运行安装程序）对话框，见图 2.3。


Add/Remove Programs Properties
Install/Uninstall
Windows Setup
Startup Disk
Run Installation Program
Browse...
< Back
Finish
Cancel
OK
Cancel
Apply
Start
My Computer
Control Panel
3:43 PM

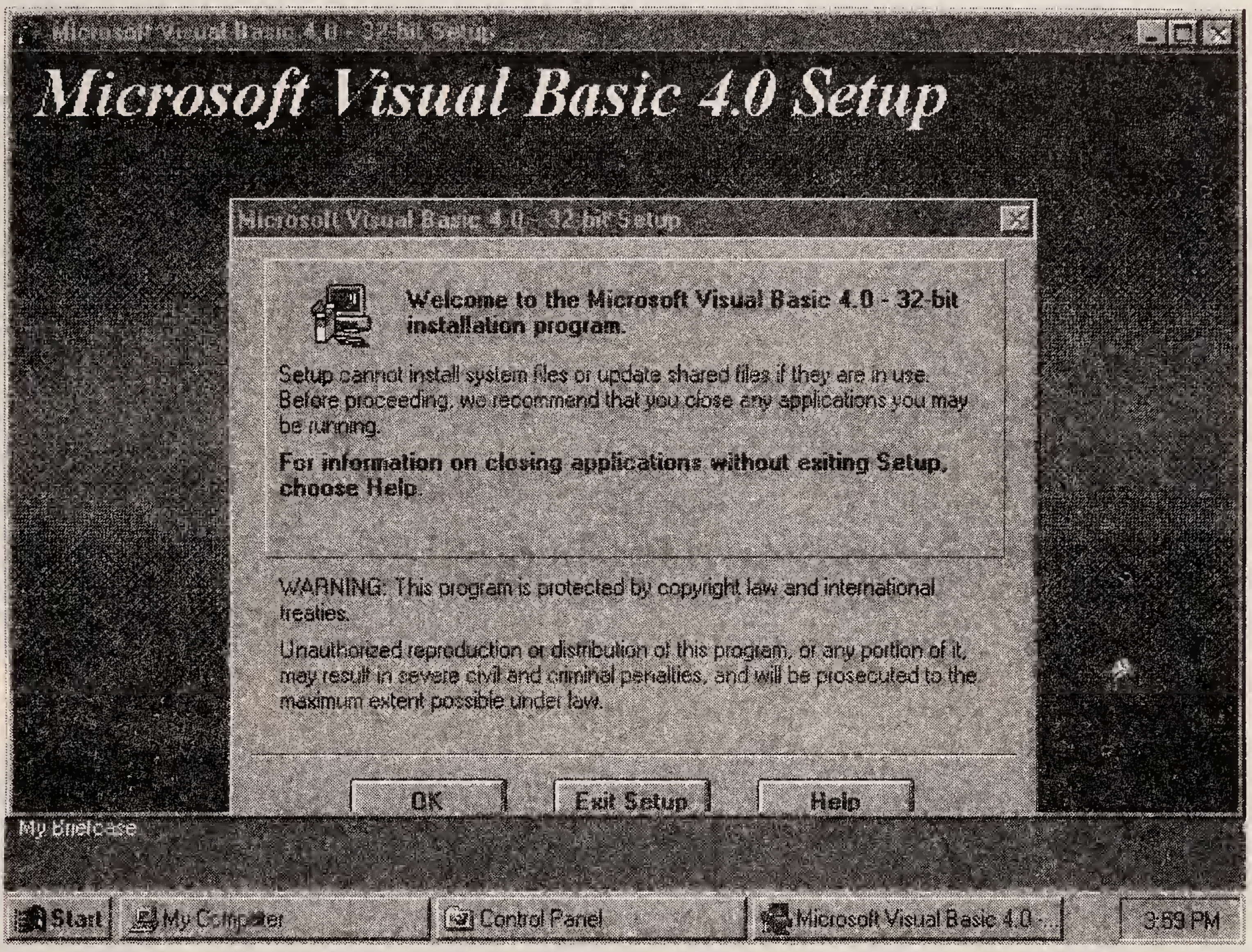
Windows was unable to locate the installation program. Click Back to start the automatic search again or click Browse to manually search for the installation program.

Command line for installation program:

图 2.3　Run Install Program（运行安装程序）对话框

用户可以在对话框的 Text 域键入安装程序的路径和文件名，此处为 C:\vb4\disk1\setup32.EXE。如果用户忘记了安装程序的路径和文件名，可以单击 Browse 按钮查阅。做好以上工作后，可以单击 Finish 按钮进行安装工作。

当用户单击 Finish 按钮后，屏幕显示如图 2.4 所示的画面。如果用户想获取有关安装 Visual Basic 4.0 的帮助信息，可以单击 Help 按钮。



Welcome to the Microsoft Visual Basic 4.0 - 32-bit installation program.

Setup cannot install system files or update shared files if they are in use. Before proceeding, we recommend that you close any applications you may be running.

For information on closing applications without exiting Setup, choose Help.

WARNING: This program is protected by copyright law and international treaties.

Unauthorized reproduction or distribution of this program, or any portion of it, may result in severe civil and criminal penalties, and will be prosecuted to the maximum extent possible under law.

Microsoft Visual Basic 4.0 - 32-bit Setup
Microsoft Visual Basic 4.0 Setup
Microsoft Visual Basic 4.0 - 32-bit Setup
OK
Exit Setup
Help
My Briefcase
Start
My Computer
Control Panel
Microsoft Visual Basic 4.0 -
3:59 PM

图 2.4　安装时的第一幅画面

单击 OK 按钮继续安装工作直到显示如图 2.5 所示的画面，这时用户可以改变 Visual Basic 4.0 应用程序存放的文件夹，缺省文件夹为\VB。单击 Change 按钮可以按用户的要求修改文件夹。

单击 OK 继续安装工作，这时系统开始寻找已安装的内容，完成以后，屏幕上显示一窗口要求用户选择安装类型，见图 2.6。有两种安装类型：

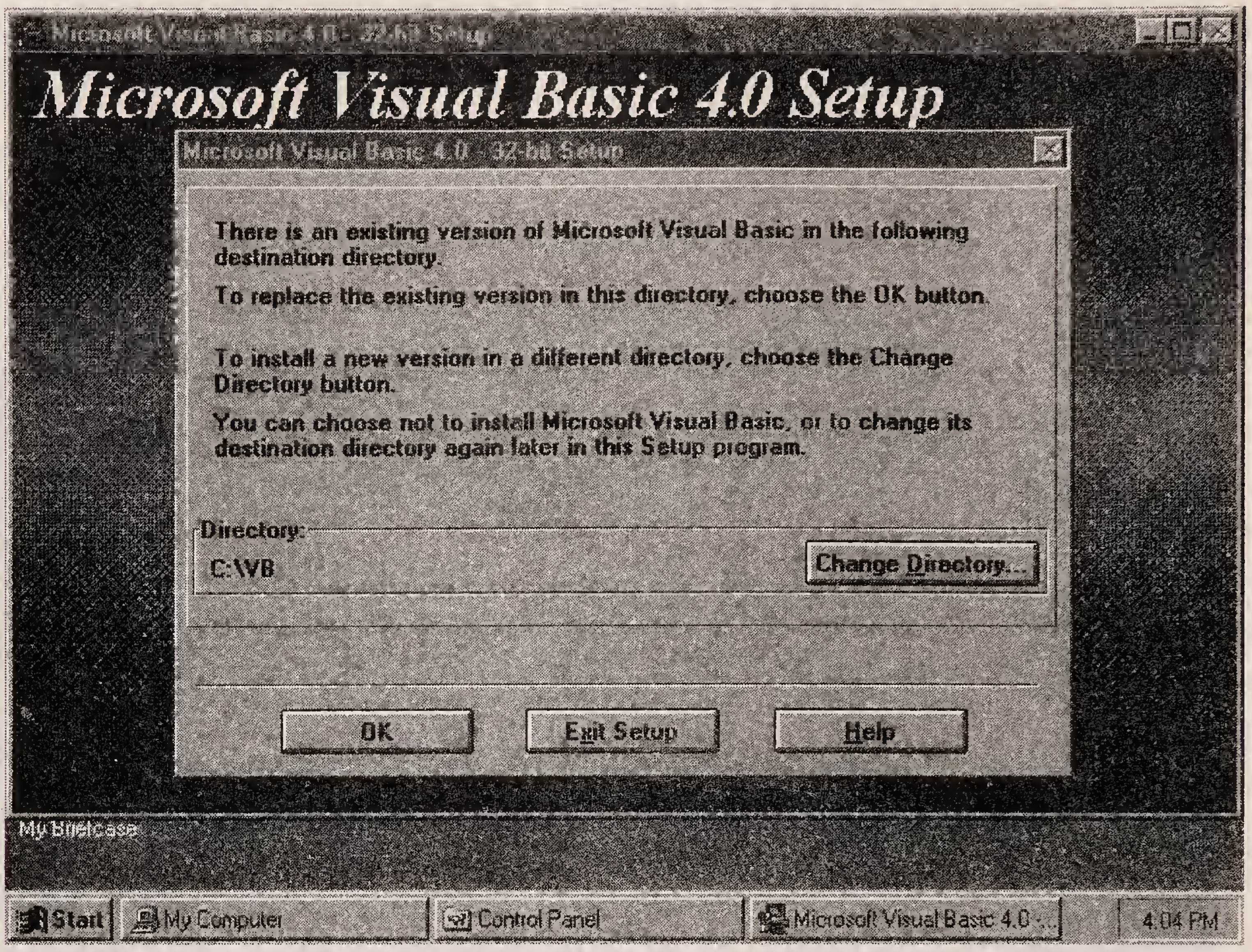
(1) Compelete(完全安装)——将 Visual Basic 的所有内容装在硬盘上

(2) Custom(定制安装)——按照用户的选择意见安装 Visual Basic 4.0

在这里，用户可以选择完全安装类型。单击完全安装类型的图标后，稍等一会，屏幕显示如图 2.7 所示的画面，它用来指定把 Visual Basic 4.0 应用程序添加到 Windows 95 Start 菜单的 Programs 的那个程序组中。

单击 Continue 按钮，选择缺省的 Visual Basic 4.0 程序组。这时安装程序系统检查是否有足够的硬盘空间，如果有足够的硬盘空间，就开始将 Visual Basic 4.0 内容拷贝到硬盘上，并且显示安装的进程，见图 2.8。

在这以后，用户可能要等一段时间。当安装进程显示为 100% 时，稍后一段时间，屏幕显示如图 2.9 所示的画面，单击 OK 完成安装工作。


Microsoft Visual Basic 4.0 - 32-bit Setup
Microsoft Visual Basic 4.0 Setup
Microsoft Visual Basic 4.0 - 32-bit Setup
Change Directory...
OK
Exit Setup
Help
My Briefcase
Start
My Computer
Control Panel
Microsoft Visual Basic 4.0 ...
4:04 PM

There is an existing version of Microsoft Visual Basic in the following destination directory.

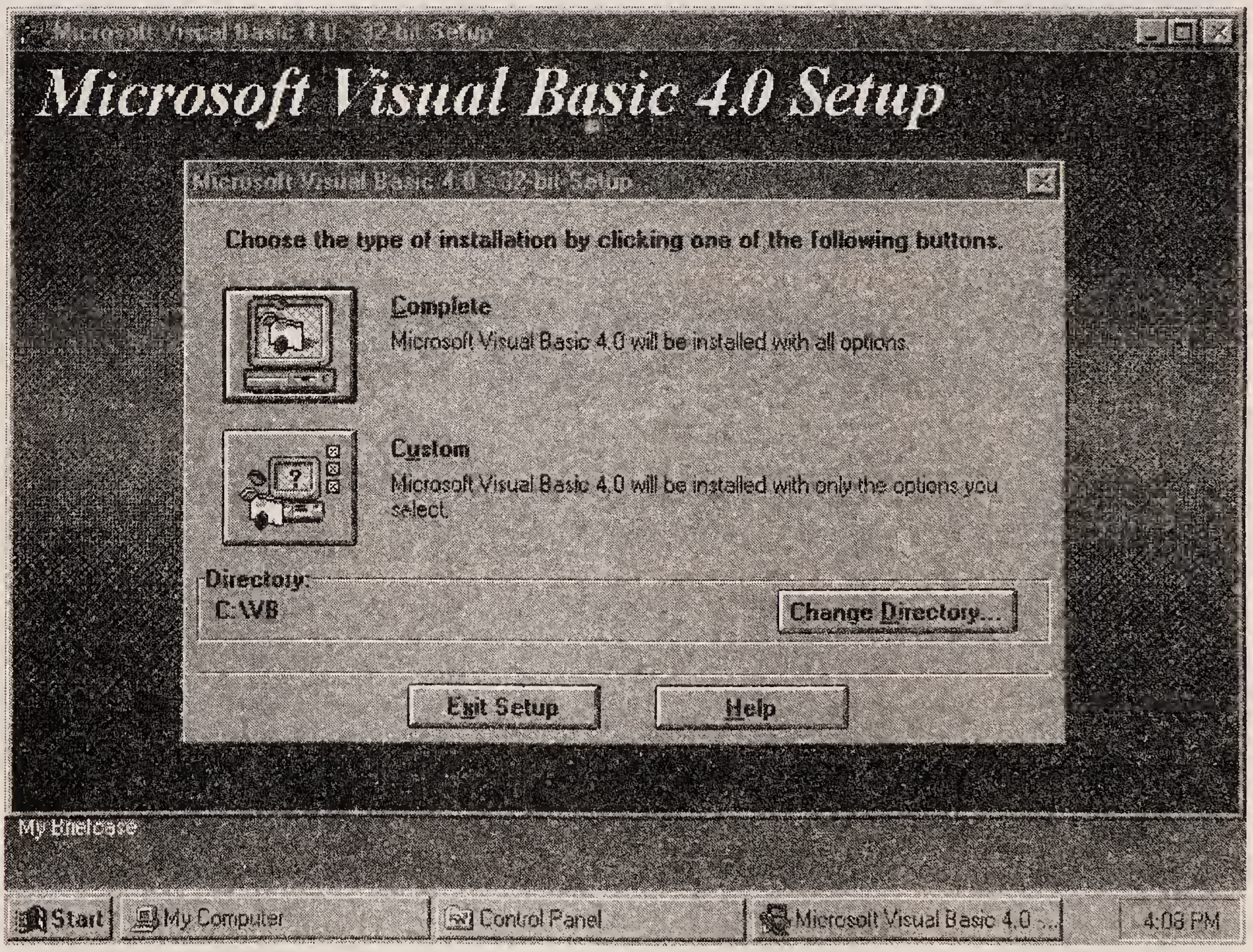
To replace the existing version in this directory, choose the OK button.

To install a new version in a different directory, choose the Change Directory button.

You can choose not to install Microsoft Visual Basic, or to change its destination directory again later in this Setup program.

Directory:
C:\VB

图 2.5　确定 Visual Basic 4.0 存放的文件夹


Microsoft Visual Basic 4.0 - 32-bit Setup
Microsoft Visual Basic 4.0 Setup
Microsoft Visual Basic 4.0 - 32-bit Setup
Complete
Custom
Change Directory...
Exit Setup
Help
My Briefcase
Start
My Computer
Control Panel
Microsoft Visual Basic 4.0 ...
4:08 PM

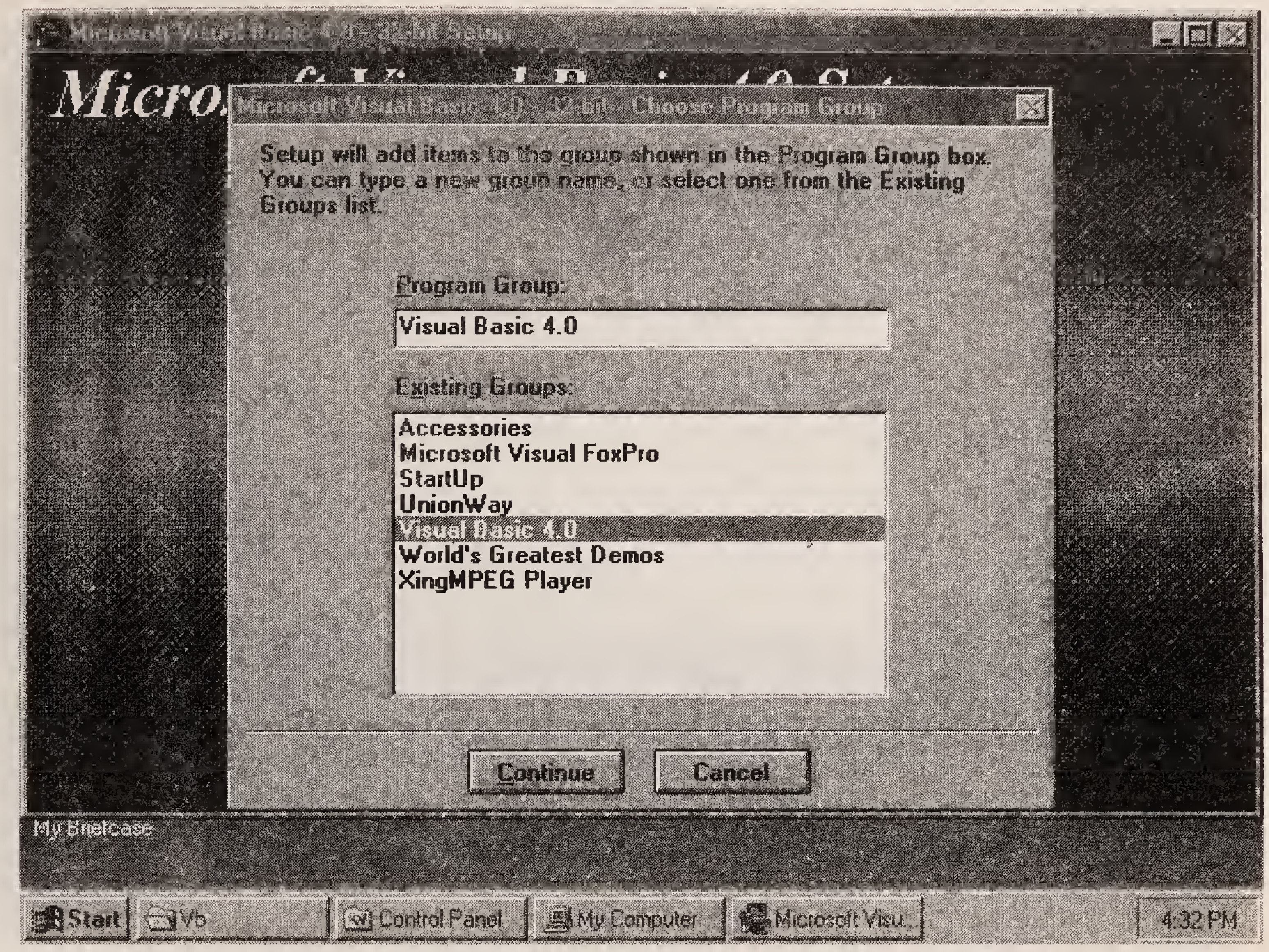
Choose the type of installation by clicking one of the following buttons.

Complete
Microsoft Visual Basic 4.0 will be installed with all options.

Custom
Microsoft Visual Basic 4.0 will be installed with only the options you select.

Directory:
C:\VB

图 2.6　选择安装类型



Microsoft Visual Basic 4.0 - 32-bit Setup

Microsoft Visual Basic 4.0 - 32-bit - Choose Program Group

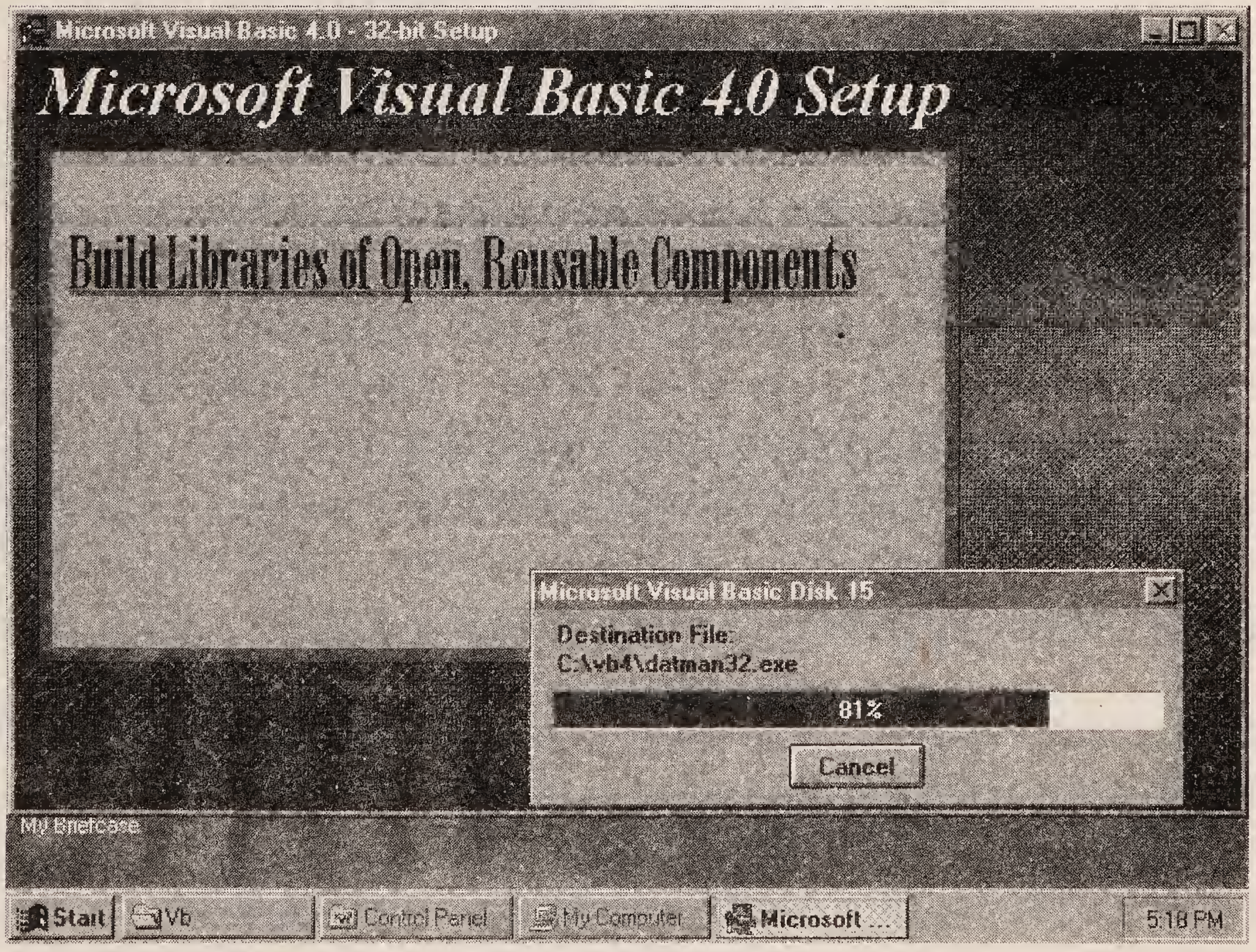
Setup will add items to the group shown in the Program Group box. You can type a new group name, or select one from the Existing Groups list.

Program Group: Visual Basic 4.0

Existing Groups:
Accessories
Microsoft Visual FoxPro
StartUp
UnionWay
Visual Basic 4.0
World's Greatest Demos
XingMPEG Player

Continue
Cancel
My Briefcase
Start
Vb
Control Panel
My Computer
Microsoft Visu...
4:32 PM

图 2.7　把 Visual Basic 4.0 程序添加到 Programs 中



Microsoft Visual Basic 4.0 - 32-bit Setup

Microsoft Visual Basic 4.0 Setup

Build Libraries of Open, Reusable Components

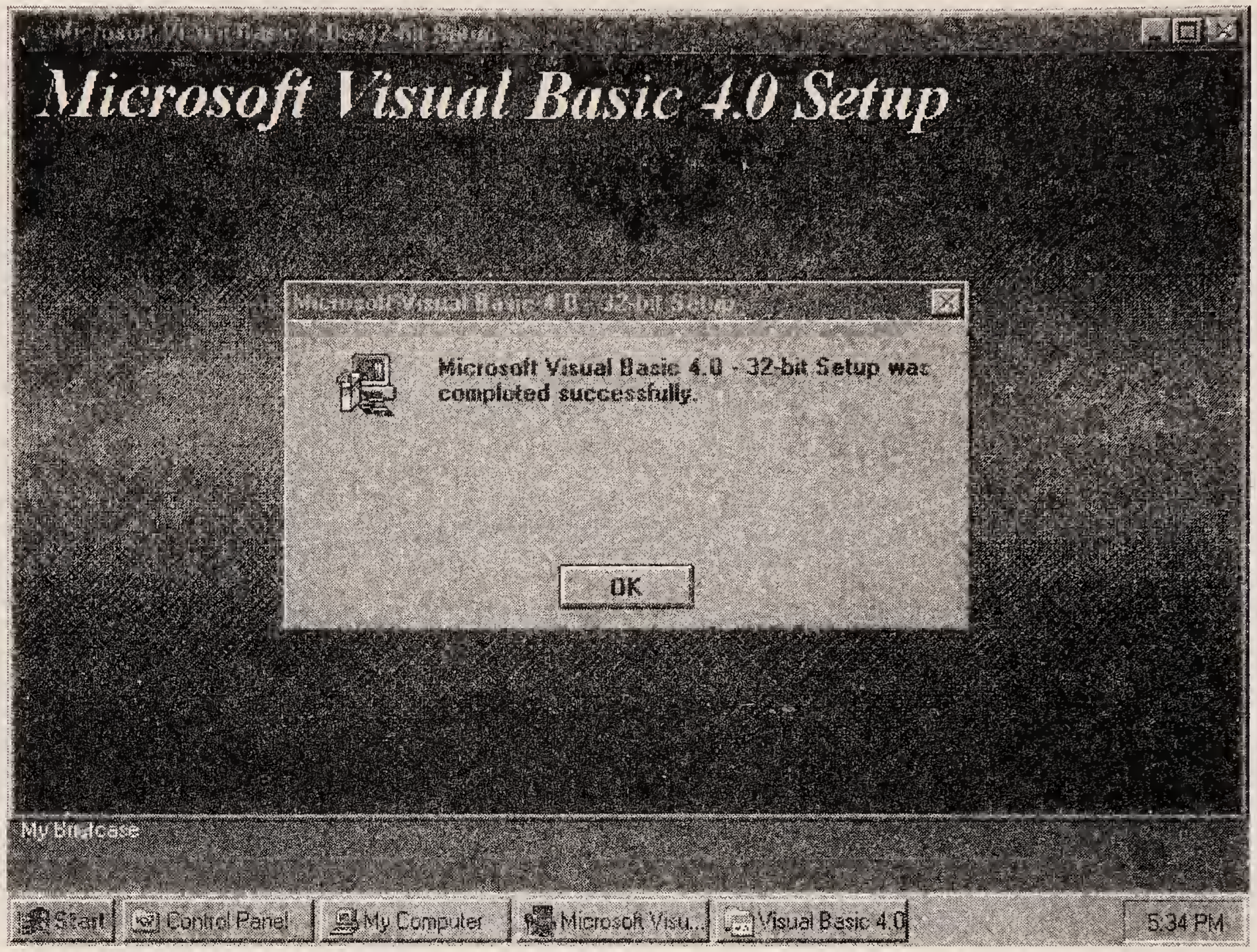
Microsoft Visual Basic Disk 15

Destination File:
C:\vb4\datman32.exe

81%

Cancel
My Briefcase
Start
Vb
Control Panel
My Computer
Microsoft ...
5:18 PM

图 2.8　安装进程显示


Microsoft Visual Basic 4.0 Setup
Microsoft Visual Basic 4.0 - 32-bit Setup
OK
My Briefcase
Start
Control Panel
My Computer
Microsoft Visu...
Visual Basic 4.0
5:34 PM

Microsoft Visual Basic 4.0 - 32-bit Setup was completed successfully.

图 2.9　向用户显示安装成功的信息

2.5　Visual Basic 4.0 的不同版本

Visual Basic 4.0 有三种不同的版本：标准版本、专业版本以及商业版本。这三种版本都是在相同的基础上建立起来的，因此许多程序在这三种版本中是通用的。Visual Basic 4.0 标准版本为一基础版本，在 Windows 95 以及 Windows NT 下完全支持 32 位应用程序。专业版本则包括与 Microsoft Jet Data Access Engine 相关的一些附加的特征，创建 OLE 自动服务器，以及与 Windows 3.1 版本中的 16 位应用程序相互兼容。它还包括 Windows 中大量的元文件以及位图，使用户可以很方便的为应用程序添加图形。Visual Basic 4.0 商业版本则是更高一层的为客户服务的 Visual Basic，它除了有 Visual Basic 4.0 专业版中所有的特征以外，还包括一些特殊的部件，用来支持高度自动化、代码检索以及提供快速数据存取的源控件等等。

如果用户是在 Windows NT 下运行 Visual Basic，可以检阅 Windows NT 用户手册来看看怎样在 Windows NT 环境下运行 Windows 应用程序。

2.6　用 Visual Basic 建立 32 位应用程序

Visual Basic 4.0 是专门为 Microsoft 的 32 位操作系统编写的、用来建立 32 位应用程序的最新 Visual Basic 版本。Windows 95 以及 Windows NT 的 32 位应用程序由于 Windows 95 和 Windows NT 的更大的可利用地址空间以及存储器，变得运行速度更快，保护性更强，并且比传统的 16 位应用程序更容易工作于多任务环境下。

在 Windows 95 或 Windows NT 环境下编译 Visual Basic 程序时，由于是由 32 位编译器编译的，因此可以自动生成 32 位应用程序。

在 Windows 95 或 Windows NT 下的 32 位 Visual Basic 应用程序有下列优点：

(1) 可以存取 Intel 386，486 或奔腾(Pentium)微处理器中的 32 位地址空间。一个 32 位的 Visual Basic 应用程序可以使用 2 千兆字节的系统存储器。

(2) 由于进行 32 位计算及存储操作，Visual Basic 性能加强了。

(3) 对于由于其他程序运行不当而造成的系统崩溃具有更高的保护功能。每一个 Visual Basic 应用程序都运行在自己的保护存储区域中，以不受其他程序中错误的影响。

(4) 可以工作在多任务工作环境中，并可以在多任务时候在不同的程序间进行切换。用户可以随时进入或退出某一 Visual Basic 应用程序，并拥有更多的控制权。

(5) 通过 Windows 32 API，用户可以为程序增加许多更高的能力。

为了与以前的 Windows 版本相互兼容，Visual Basic 4.0 专业版本中包括一个独立的 Visual Basic 4.0 16 位版本，允许用户对 Windows 3.1 版本下的 16 位应用程序进行编译。用户用 Visual Basic 4.0 16 位版本建立的工程文件与 32 位是完全兼容的。这就使用户可以建立一个适用于当前所有版本的应用程序，这也是为了能让用户运行以前 Visual Basic 版本中的应用程序着想的。

2.7　怎样使用 Visual Basic 的帮助系统

用户初学 Visual Basic，少不了要了解 Visual Basic 的许多内容，这就得借助 Visual Basic 的帮助系统。

Visual Basic 4.0 为用户提供了大量的在线帮助信息。包括编程环境、工具、编程语言等等。寻求帮助可以通过好几种方式。

(1) 单击 Visual Basic Help 菜单中的 Contents 命令，即出现帮助题目以及内容目录，如图 2.10 所示。

(2) 在窗口中进行工作时，可以单击键盘上的 F1 键，即得到正在操作对象的帮助内容。

(3) 通过单击 Help 菜单中的 Search For Help On... 命令，即出现一对话框，键入用户想要查找的单词名即可，如图 2.11 所示。

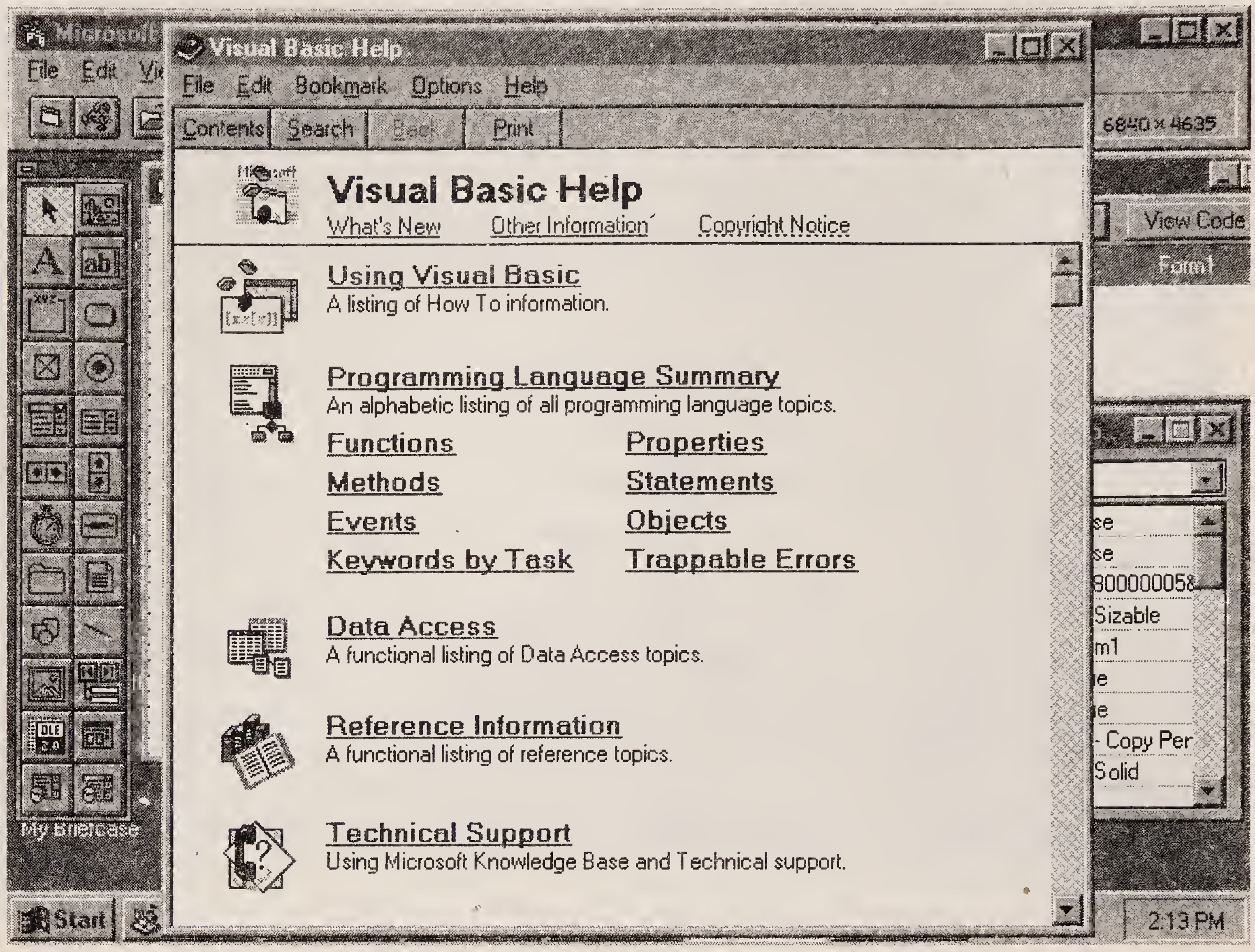
单击 Help 菜单中的 Learning Microsoft Visual Basic 命令，即出现一课程目录，通过单击某一课或单击 Instructions 即可学习每一课关于编程的内容。

下面我们就来介绍一下怎样利用 Search For Help On 对话框来寻找关于 Form Window 的帮助内容，通过这些步骤的练习，读者很快就会掌握这种寻找帮助的方法。

(1) 启动 Visual Basic，关于怎样启动，读者可以先阅读一下第三章第一节的内容。

(2) 在菜单条中单击 Help 菜单，即弹出一菜单，从中单击 Search For Help On 命令，即出现图 2.11 所示的对话框。

(3) 在文本框中键入“F”，然后键入“O”，这时在下面的列表框中出现以“FO”开头的单词，如果还没出现 Form Window，则继续往下输，直到出现为止，如图 2.12 所示。


Visual Basic Help
File
Edit
Bookmark
Options
Help
Contents
Search
Back
Print
6840 x 4635
View Code
Form1
Start
2:13 PM
My Briefcase

Visual Basic Help

What's New　Other Information　Copyright Notice

Using Visual Basic
A listing of How To information.

Programming Language Summary
An alphabetic listing of all programming language topics.

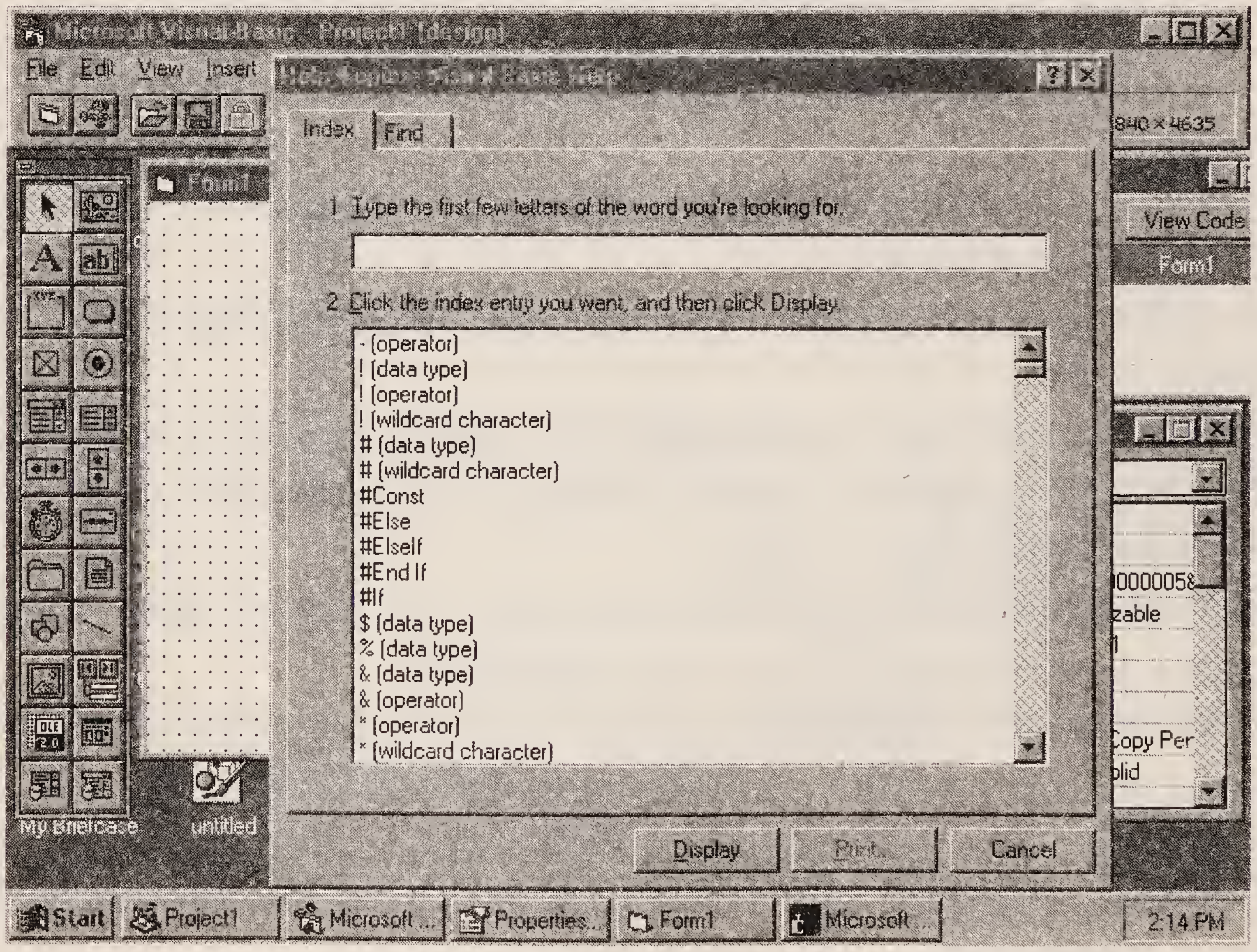
Functions　Properties
Methods　Statements
Events　Objects
Keywords by Task　Trappable Errors

Data Access
A functional listing of Data Access topics.

Reference Information
A functional listing of reference topics.

Technical Support
Using Microsoft Knowledge Base and Technical support.

图 2.10　Help 中的 Contents

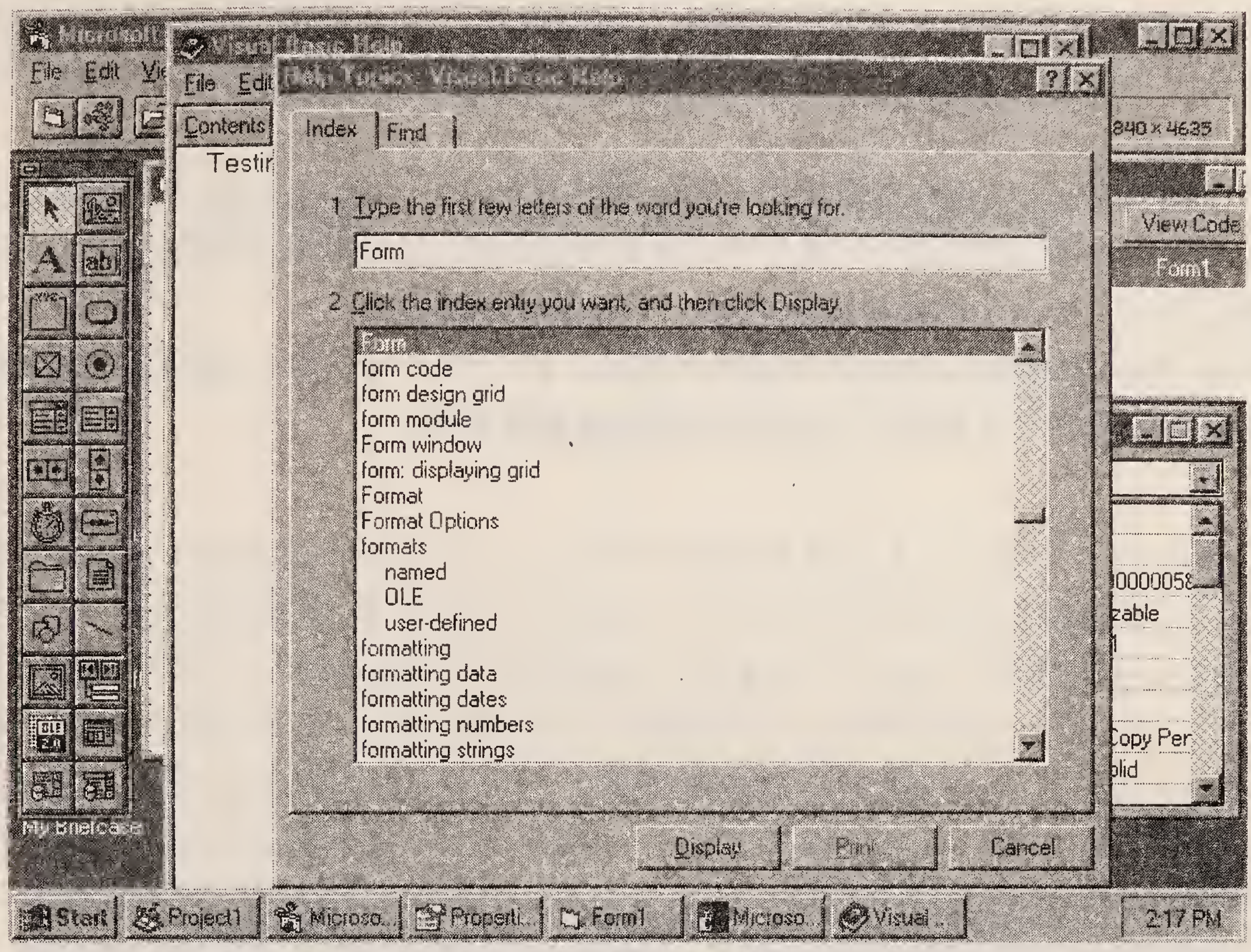

Microsoft Visual Basic - Project1 [design]
File
Edit
View
Insert
Help Topics: Visual Basic Help
Index
Find
Form1
6840 x 4635
View Code
Form1
My Briefcase
untitled
Display
Print...
Cancel
Start
Project1
Microsoft ...
Properties...
Form1
Microsoft ...
2:14 PM

1 Type the first few letters of the word you're looking for.

2 Click the index entry you want, and then click Display.

- (operator)
! (data type)
! (operator)
! (wildcard character)
(data type)
(wildcard character)
#Const
#Else
#ElseIf
#End If
#If
$ (data type)
% (data type)
& (data type)
& (operator)
* (operator)
* (wildcard character)

图 2.11　Search For Help On 对话框


Visual Basic Help
Help Topics: Visual Basic Help
File Edit
Contents
Testir
Index
Find
840 x 4635
View Code
Form1
Display
Print...
Cancel
Start
Project1
Microso...
Properti...
Form1
Microso...
Visual ...
2:17 PM

1 Type the first few letters of the word you're looking for.

Form

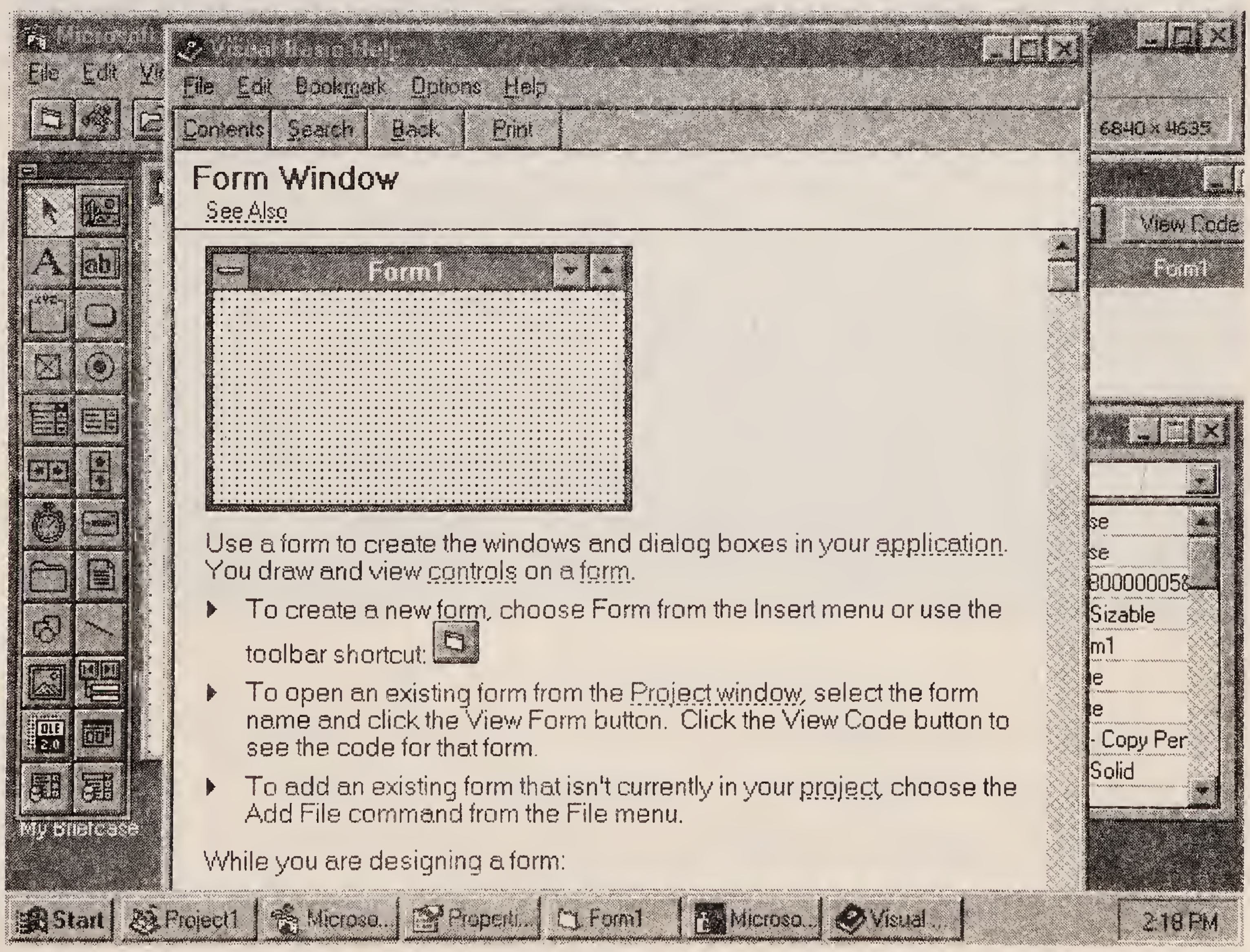
2 Click the index entry you want, and then click Display.

Form
form code
form design grid
form module
Form window
form: displaying grid
Format
Format Options
formats
  named
  OLE
  user-defined
formatting
formatting data
formatting dates
formatting numbers
formatting strings

图 2.12 在文本框中输入寻找单词的头几个字母

（4）在列表框中双击 Form Window 标题，则在上面的文本框中出现 Form Window。

（5）单击 Display 按钮，则 Help 窗口将显示有关 Visual Basic Form 窗口的有关信息，如图 2.13 所示。如果内容太多，在窗口中会出现滚动条，允许用户查看所有内容。


Visual Basic Help
File Edit Bookmark Options Help
Contents
Search
Back
Print
6840 x 4635
View Code
Form1
See Also
Form1
Sizable
Copy Per
Solid
Start
Project1
Microso...
Properti...
Form1
Microso...
Visual ...
2:18 PM

Form Window

Use a form to create the windows and dialog boxes in your application. You draw and view controls on a form.

- To create a new form, choose Form from the Insert menu or use the toolbar shortcut:
- To open an existing form from the Project window, select the form name and click the View Form button. Click the View Code button to see the code for that form.
- To add an existing form that isn't currently in your project, choose the Add File command from the File menu.

While you are designing a form:

图 2.13 Form 窗口帮助信息

(6) 在窗口的标题栏下会有一有绿色下划线的 See Also 选项，单击它时，会弹出一窗口，单击窗口中的感兴趣的与 Form 窗口有相关内容的题目，即出现这一题目所包含的帮助信息。

(7) 通过单击 ESC 键即可退出 See Also 弹出窗口。

(8) 单击 File 菜单中的 Exit 命令即可退出帮助系统。也可以单击 Help 窗口标题条中的关闭图标，即右上角三个图标最右边的那一个来退出帮助系统。

Visual Basic 的帮助系统是学习编程环境及了解有关 Visual Basic 编程的内容的有力助手，在碰到疑问时别忘了使用 Visual Basic 的在线帮助系统。

第三章　Visual Basic 4.0 编程环境

3.1　启动 Visual Basic 4.0

假设大家已经安装好 Visual Basic 4.0，那么现在我们就来启动它。首先进入 Windows 95 桌面系统，在前面我们已对 Windows 95 做了简单的介绍，这里就直接使用它。大家会看到在桌面的左下角有一 Start 按钮，把指针移到上面单击一下鼠标，在其右边会弹出一菜单。再把光标移到 Programs 上，其右边会弹出一级联菜单。把光标移到 Visual Basic 4.0 上，其右边会出现 Visual Basic 4.0 程序组，然后我们把光标移到 Microsoft Visual Basic 4.0-32-bit，并单击鼠标左按钮，即可进入 Visual Basic 4.0，其过程如图 3.1 所示。

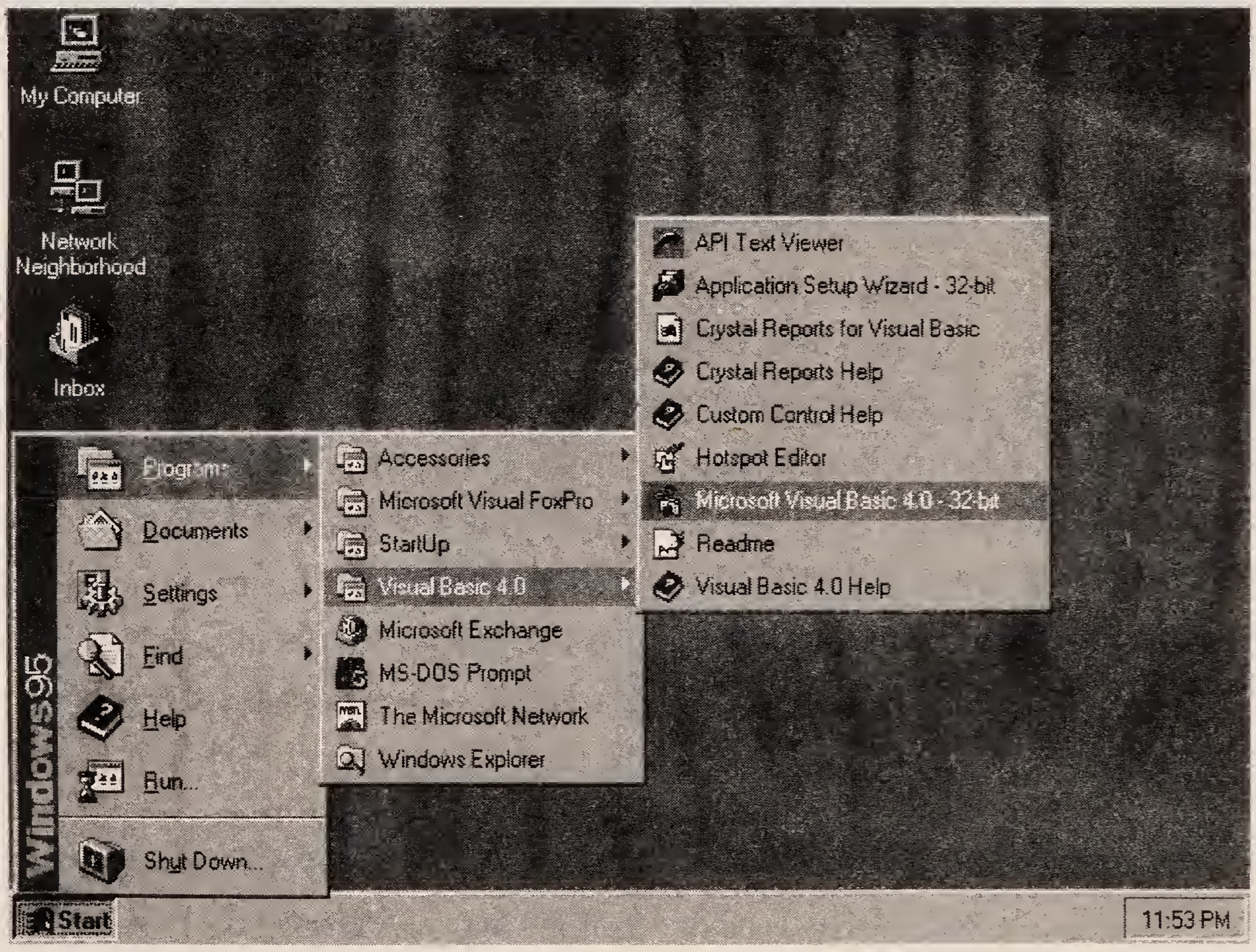

My Computer
Network Neighborhood
Inbox
Programs
Documents
Settings
Find
Help
Run...
Shut Down...
Windows 95
Accessories
Microsoft Visual FoxPro
StartUp
Visual Basic 4.0
Microsoft Exchange
MS-DOS Prompt
The Microsoft Network
Windows Explorer
API Text Viewer
Application Setup Wizard - 32-bit
Crystal Reports for Visual Basic
Crystal Reports Help
Custom Control Help
Hotspot Editor
Microsoft Visual Basic 4.0 - 32-bit
Readme
Visual Basic 4.0 Help
Start
11:53 PM

图 3.1　从 Start 启动 Visual Basic

第二种方法是把鼠标指针移到 My Computer（我的电脑），然后双击鼠标，即出现一窗口，我们选择 Disk[C:]，即可进入 C 盘驱动器窗口，C 盘驱动器窗口内为 C 盘驱动器的文件和文件夹。再找到 vb 文件夹并打开它，找到 vb 32 图标并双击鼠标。即可进入 Visual Basic 4.0，其过程如图 3.2 所示。

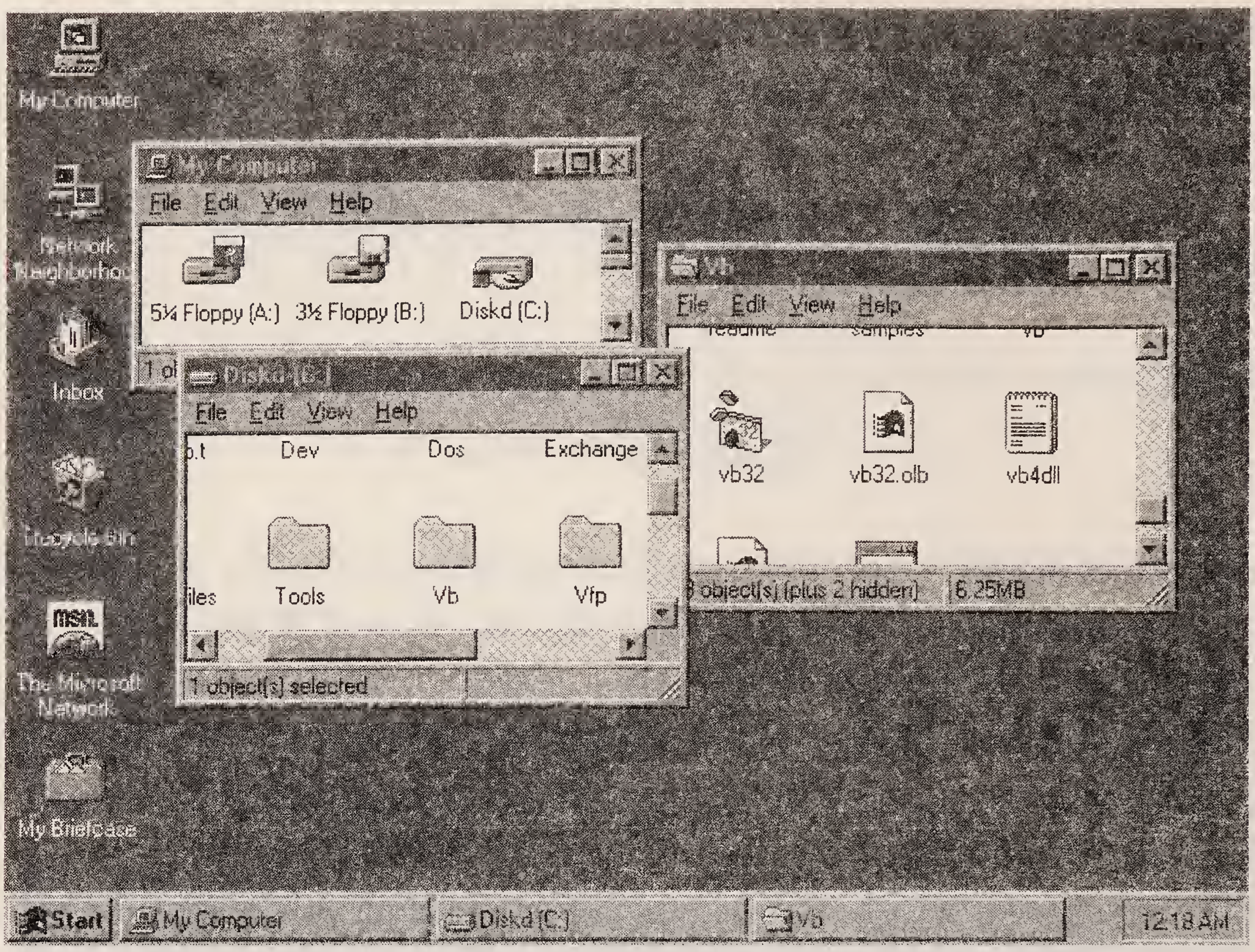

My Computer
Network Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
My Computer
File Edit View Help
5¼ Floppy (A:)
3½ Floppy (B:)
Diskd (C:)
Diskd (C:)
File Edit View Help
Dev
Dos
Exchange
Tools
Vb
Vfp
1 object(s) selected
Vb
File Edit View Help
readme
samples
vb
vb32
vb32.olb
vb4dll
8 object(s) (plus 2 hidden)
6.25MB
Start
My Computer
Diskd (C:)
Vb
12:18 AM

图 3.2　从 My Computer 启动 Visual Basic

3.2　Visual Basic 的用户环境

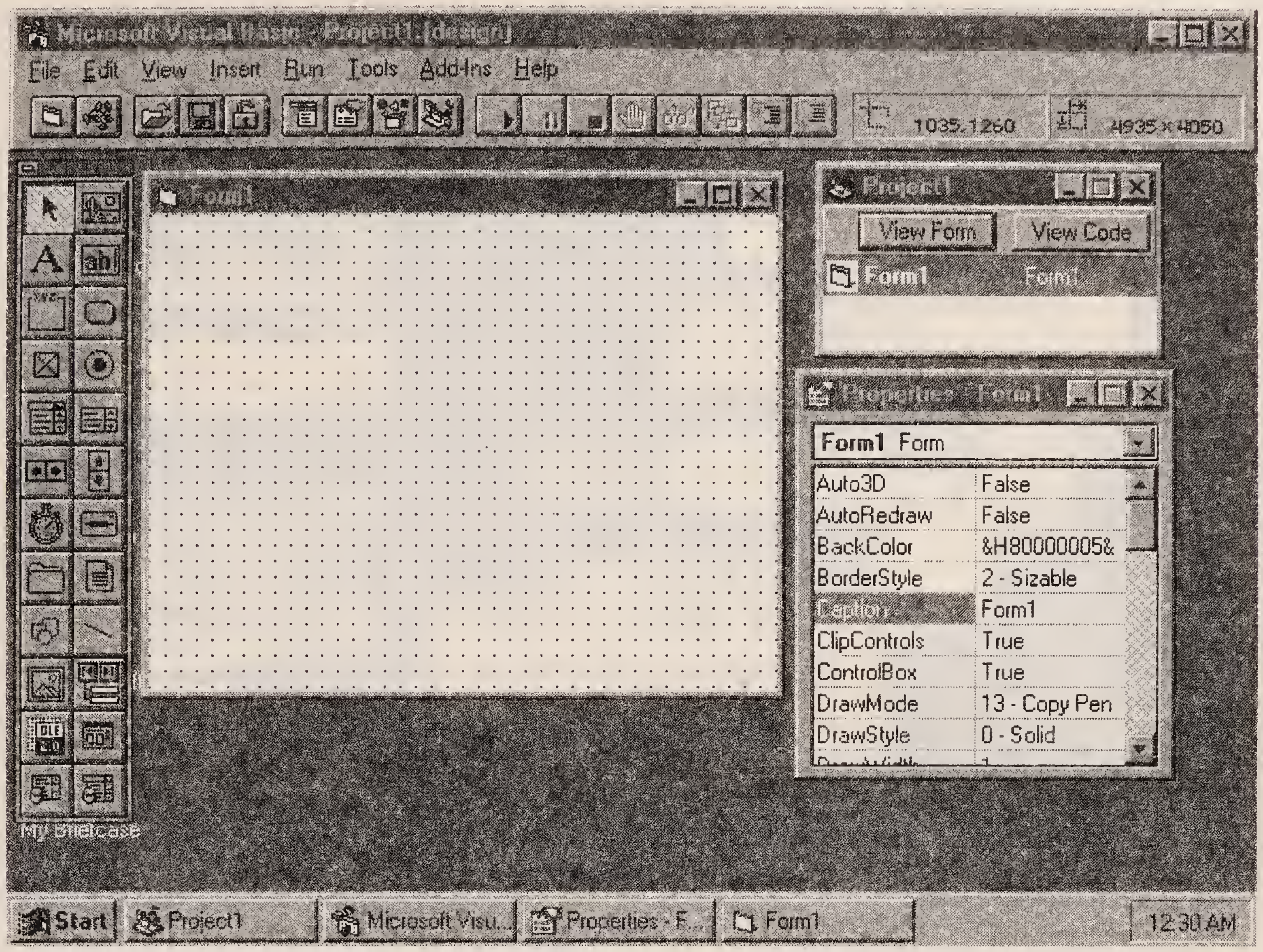
启动 Visual Basic 4.0 以后在屏幕上出现如图 3.3 所示的画面，这就是我们进行操作的图形界面。在屏幕上我们能看到的只有四个窗口和标题条(Title bar)、菜单条(Menu bar)以及工具栏(Toolbar)。在整个屏幕的下方有几个按钮，它们分别是 Start(开始)，Project1(工程文件 1)，Microsoft Visual Basic 4.0(主窗口)，properties-Form1(Form1 属性窗口)，Form1(窗体 1 窗口)。表示成被按下去形状的按钮表示其所代表的窗口是当前活动窗口，这些按钮为用户在各个窗口间进行切换提供了方便，当然我们要想激活某一窗口，也可以对着这一窗口单击一下鼠标即可。

3.2.1　Visual Basic 各个窗口的作用

Visual Basic 启动以后在屏幕上就打开了四个窗口，它们分别是 Form(窗体)窗口，Project(工程文件)窗口，屏幕最左面的 Toolbox(工具箱)窗口，以及 Properties(属性)窗口。另外还有几个窗口是隐形的，它们分别是 Color Palette(调色板)窗口，Code(代码)窗口，Debug(调试)窗口，Menu Editor(菜单编辑)窗口，Object Browser(对象浏览器)窗口，下面我们就逐一进行介绍。

(1) Form(窗体)窗口

Form 窗口，我们可以在图 3.3 的中间位置看到它。Form 窗口是生成用户应用程序的编辑窗口和对话框。用户可在一个应用程序中拥有多个 Form 窗口，每个 Form 窗口都有不同的名字，以免运行时发生混淆。在缺省时 Form 分别被命名为 Form1，Form2，Form3...，如


Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
1035,1260
4935 x 4050
Form1
Project1
View Form
View Code
Form1 Form1
Properties - Form1
My Briefcase
Start
Project1
Microsoft Visu...
Properties - F...
Form1
12:30 AM

Form1 Form

Auto3D	False
AutoRedraw	False
BackColor	&H80000005&
BorderStyle	2 - Sizable
Caption	Form1
ClipControls	True
ControlBox	True
DrawMode	13 - Copy Pen
DrawStyle	0 - Solid

图 3.3 Visual Basic 的工作环境

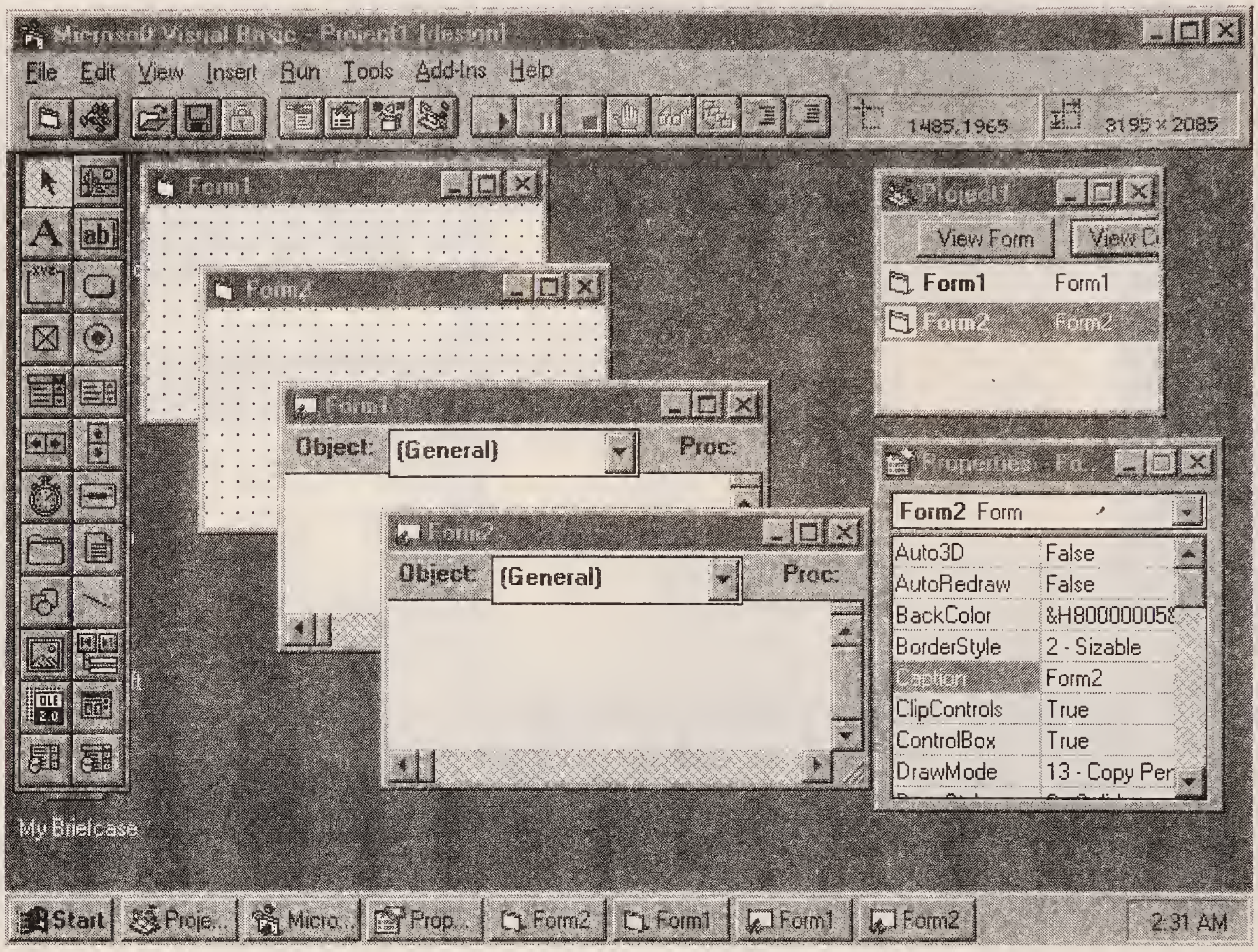
果用户不想用这些缺省名字，可以自己创建一些新名字，这样便于用户理解每个 Form 的功能，更容易进行跟踪。在设计应用程序时，所有的 Form 都是可视的，但用户所设计的程序则可以把这些 Form 窗口设为可视的或不可视的，这要根据具体程序而定。

(2) Project(工程文件)窗口

工程文件窗口在图 3.3 的右上方，它保存有一个应用程序的所有属性以及组成这个应用程序的所有文件，在工程文件窗口下面的空白处将列出这个应用程序的所有文件。在标题条 Project 下面有左右两个按钮。左面的 View Form 按钮表示可以通过它回到一个 Form 窗口。而右边的 View Code 按钮则表示当用户按下它时，可以打开一个 Code 窗口。例如我们创建两个 Form 窗口，在 Project 窗口的下面空白列表框部分将会出现 Form1 和 Form2 两个文件。当我们选择 Form1 并单击 View Form 时，则会在屏幕中央出现 Form1 窗口，当选择 Form2 并单击 View Form 时，即出现 Form2 窗口。当选择 Form1 并单击 View Code 时则出现 Form1 代码窗口，如图 3.4 所示。并且当前选中的窗口为活动窗口，这样用户就可以很方便的选用、切换窗口。当选中一个 Form 或 Code 窗口后，我们就可以对它进行编辑了。

(3) Properties(属性)窗口

属性窗口在图 3.3 的右下角。所有的窗体都有颜色、字体、大小、尺寸之类的属性，对于这些属性，我们可以通过属性窗口来对窗体进行修改。属性窗口的最上面是标题条，中间是一文本框，里面列出了当前选定的活动对象及其名称，文本框的右边是一向下的箭头，当用鼠标单击这一箭头时，将在其下方出现一列表框，列出了 Form1 窗口的所有对象及其名称。当激活某一对象时，这一对象所在的行将变为高亮。当然我们也可以通过在列表里选定某一对象来激活它，如图 3.5 所示。在文本框的下面是一较大的列表框，列出了用户选定对象的各种属性，在这里要知道，对于不同对象它所列出的属性也是不同的。属性列表由中间一条

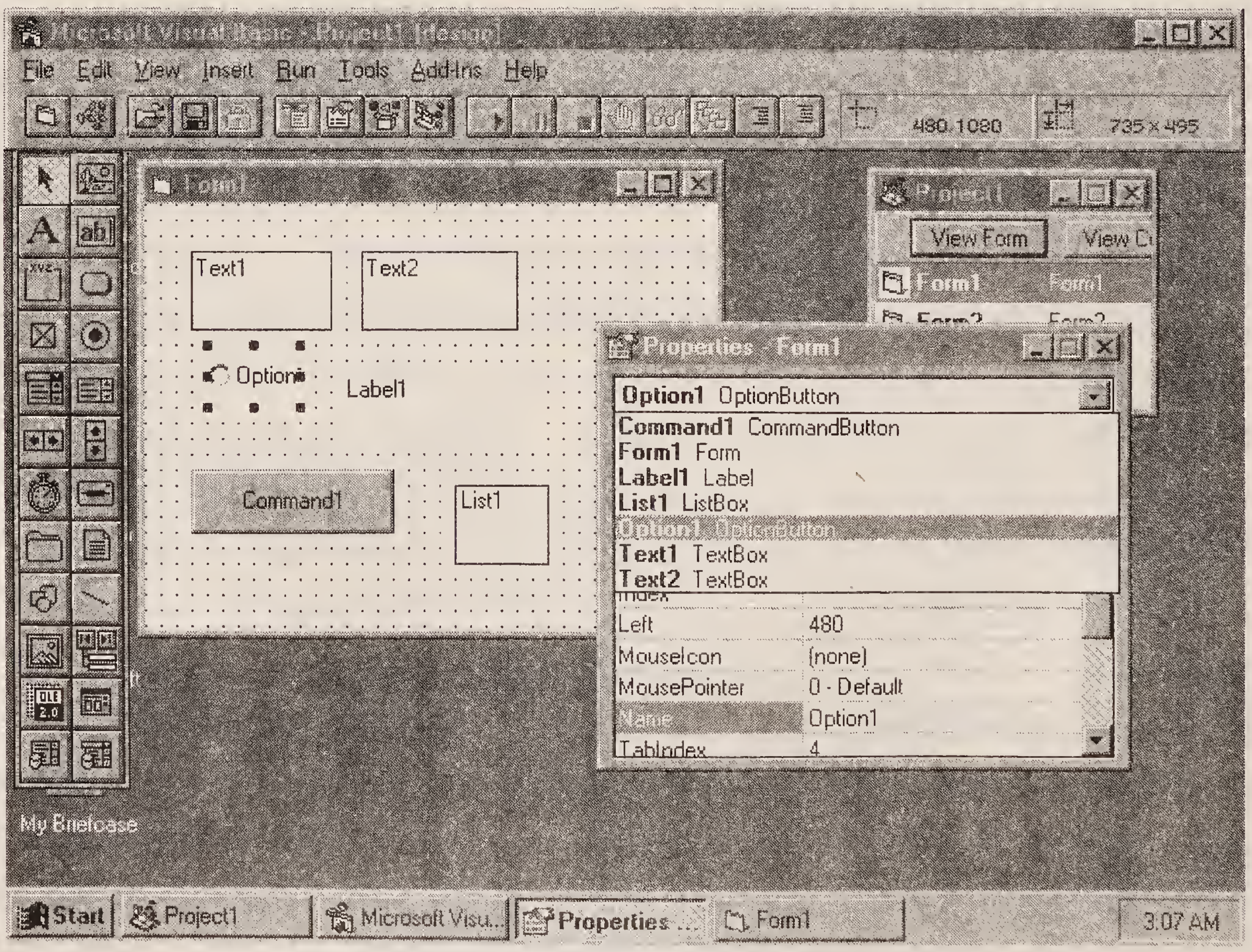

Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
1485,1965
3195 x 2085
Form1
Form2
Project1
View Form
Form1 Form1
Form2 Form2
Object: (General)
Proc:
Properties - Fo...
My Briefcase
Start Proje... Micro... Prop... Form2 Form1 Form1 Form2
2:31 AM

Form2 Form

Auto3D	False
AutoRedraw	False
BackColor	&H80000005
BorderStyle	2 - Sizable
Caption	Form2
ClipControls	True
ControlBox	True
DrawMode	13 - Copy Pen

图 3.4　Project 窗口显示的当前工程文件中的文件列表

线把其分为两部分，左面列出的是各种属性，右面列出的则是相应的属性值。当选定某一属性时，将光标移到右面对应的设置属性值区，单击鼠标，将出现一活动光标，如果想保留缺省值，则不必改动。如想改变设置，则将原缺省值删掉，键入新设置的值，回车即可。


Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
480,1080
735 x 495
Form1
Text1
Text2
Option1
Label1
Command1
List1
Project1
View Form
Form1 Form1
Form2 Form2
Properties - Form1
My Briefcase
Start Project1 Microsoft Visu... Properties ... Form1
3:07 AM

Option1 OptionButton

Command1 CommandButton
Form1 Form
Label1 Label
List1 ListBox
Option1 OptionButton
Text1 TextBox
Text2 TextBox

Left	480
MouseIcon	(none)
MousePointer	0 - Default
Name	Option1
TabIndex	4

图 3.5　Properties 窗口显示选中的控件及窗体属性

(4) Toolbox（工具框）窗口

Toolbox 窗口位于图 3.3 的最左边，由 24 个被绘成按钮形式的图标所构成。Toolbox 里的所有对象都可以置于 Form 窗口中。每一个对象都是一个真实的实物及对这个实物进行操作的代码的抽象组合体。也就是说，一个对象在屏幕上具有某种形状，或至少在某个应用程序中对某个物理形状要有所响应。下面我们就对每个图标所表示的 Toolbox 窗口中的对象逐一介绍，其中最后三个是 Visual Basic 4.0 版本中新增添的。

Printer（指针）——这是在 Toolbox 中唯一没有控制功能的对象，我们可以在某一控件被加到 Form 中后来移动它以及改变它的大小。当选定指针对象后，只能用来移动控件及改变它的尺寸大小。当选定其他控件时，只能用它来移动新选定的控件或改变它的大小。当用户选定某一控件后，指针将被自动选定。

Picture box（图片框）——用来显示一幅图画，也可以作为一个容器，用来接收图片方法(graphics method)的输出，或作为其他控件的载体。

Lable（标签）——用来显示不想让其他用户改变的文本标签。例如，一幅图画的标题。

Textbox（文本框）——用来显示用户可以进行编辑的文本。

Frame（框架）——用来建立一组可视的功能框架。如建立一个组合控制，首先画一框架，然后把其他控件加到里面。

Command button（命令按钮）——用来建立一个用户可以实现一种命令的按钮。

Checkbox（复选框）——用来建立一个简单的用户可以选择某项为 True（真）或 False（假）的框架，或一个拥有许多选择项的框架。

Option button（选项按钮）——用于设置一个拥有许多选项而只能选择其一的按钮。

Combobox（组合框）——用于将 Textbox（文本框）与 Listbox（列表框）组合在一起。即用户可以进入一文本框进行编辑，又可以选取列表框中的内容。

Listbox（列表框）——用来显示一个列表。如果项目太多而无法在同一时刻显示，列表框还可以将它们卷起来。

Hscrollbar(Horizontal scroll bar 水平滚动条)——可以提供一个可视的工具使用户快速的浏览一个具有很多条目的列表框或拥有大量信息的文本框，并带有一可视的水平标度尺来表明当前显示的文件内容在整个文件中的位置，或用于设置一个数值输入的设备。

Vscrollbar(Vertical scroll bar 竖直滚动条)——提供一个可视的工具使用户可以快速的浏览一个具有很多条目的列表框或有大量信息的文本框，并带有一可视的垂直标度尺来表示当前显示内容的位置，或用于设置一个数值输入的设备。

Timer（时钟）——用于在给定时刻触发某一事件的发生，在程序运行时是不可视的。

DriveListbox（驱动器列表框）——用于显示驱动器的列表框。

DirListbox(directory list box 目录列表框)——用于显示目录及路径的列表框。

FileListbox（文件列表框）——用于显示当前目录中所有文件的列表框。

Shape（形状）——在设计程序时可以在 Form 窗口中绘制大量图形，如正方形、矩形和圆形等等。

Line（直线）——在设计程序时可以在 Form 窗口中画线。

Image（图像框）——可以在 Form 窗口中显示一位图或一图标，是 Picture 框的简化版本。

Data（数据）——通过 Form 窗口的约束控制，从数据库里存取数据。

OLE Container（OLE 容器）——可以在某一应用程序中嵌入其他应用程序中的对象。

Command dialog（命令对话框）——可以建立一个能够打开、保存或打印文件的对话框。命令对话框是一定制控件。

DBlist（data-bound listbox 数据约束列表框）——可以显示一个列表框。如果内容太多，列表框还可以把它们加以滚动，数据约束列表框具有更高的数据存取能力。

DBCombo（data-bound combobox 数据约束组合框）——用于描绘一个列表框与文本框的组合体。用户既可以从列表框中选择一个项目又可以对文本框进行编辑。数据约束组合框是一个自定义控件，并且具有一种标准组合框没有的功能，那就是它具有更高的数据存取能力。

若要连接某一对象，只要将鼠标对准工具框中该对象，单击鼠标即可。这时代表该对象的按钮将被按下，同时上面的图标变为红色。当鼠标指针移到 Form 窗体中后，将变为十字。这时按下鼠标左键，并拖动鼠标，画出一方框。然后松开鼠标按钮，这时即可在 Form 窗口中产生一四周带有八个黑色小框架的方框。如图 3.6 所示。当我们把鼠标对准某一对象双击鼠标时与上面所做的步骤的结果是一样的，即马上在 Form 窗体中的缺省位置生成一带有八个黑色小方框的框架。

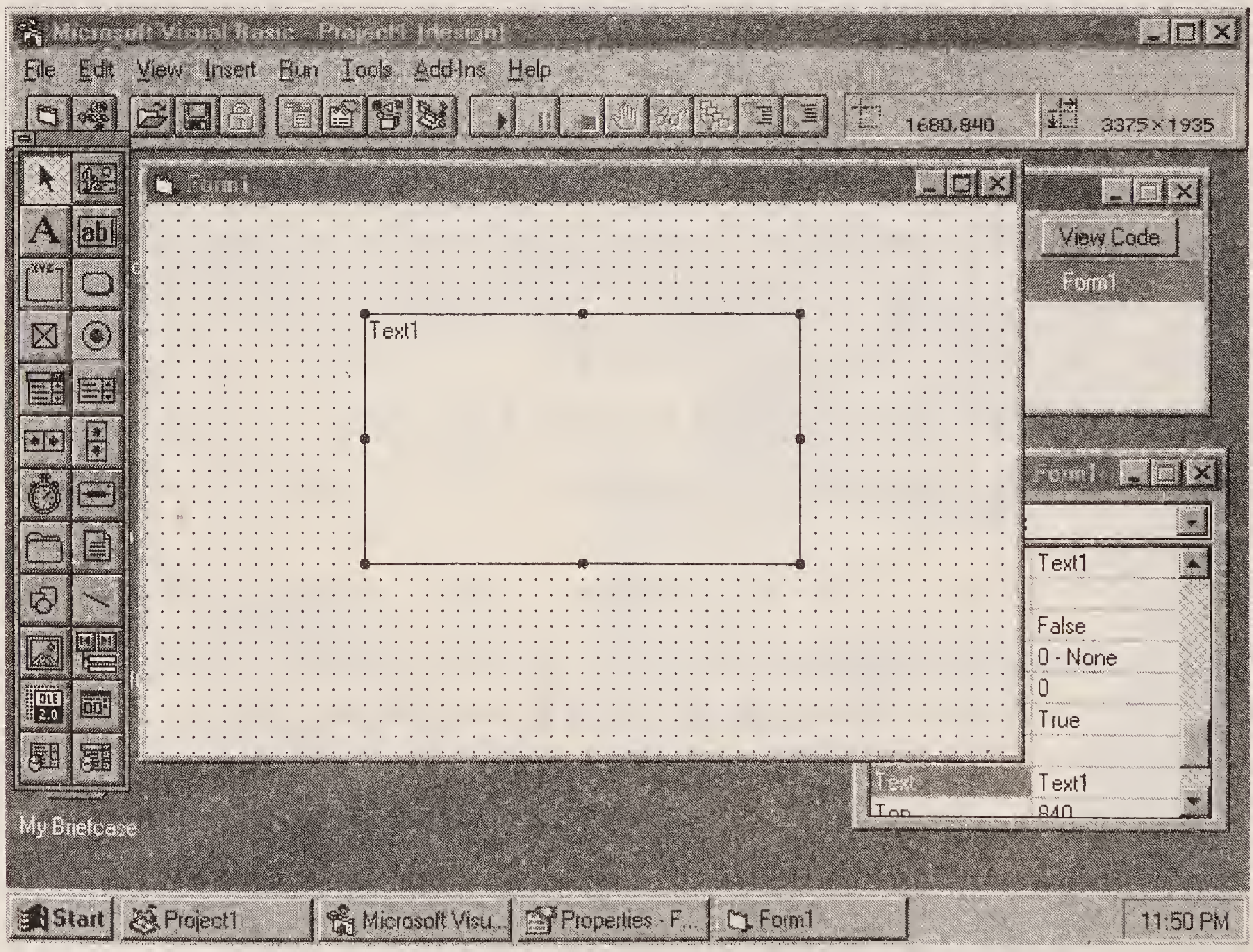

Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
1680,840
3375×1935
Form1
Text1
View Code
Form1
Form1
Text1
False
0 - None
0
True
Text
Text1
Top
840
My Briefcase
Start
Project1
Microsoft Visu...
Properties - F...
Form1
11:50 PM

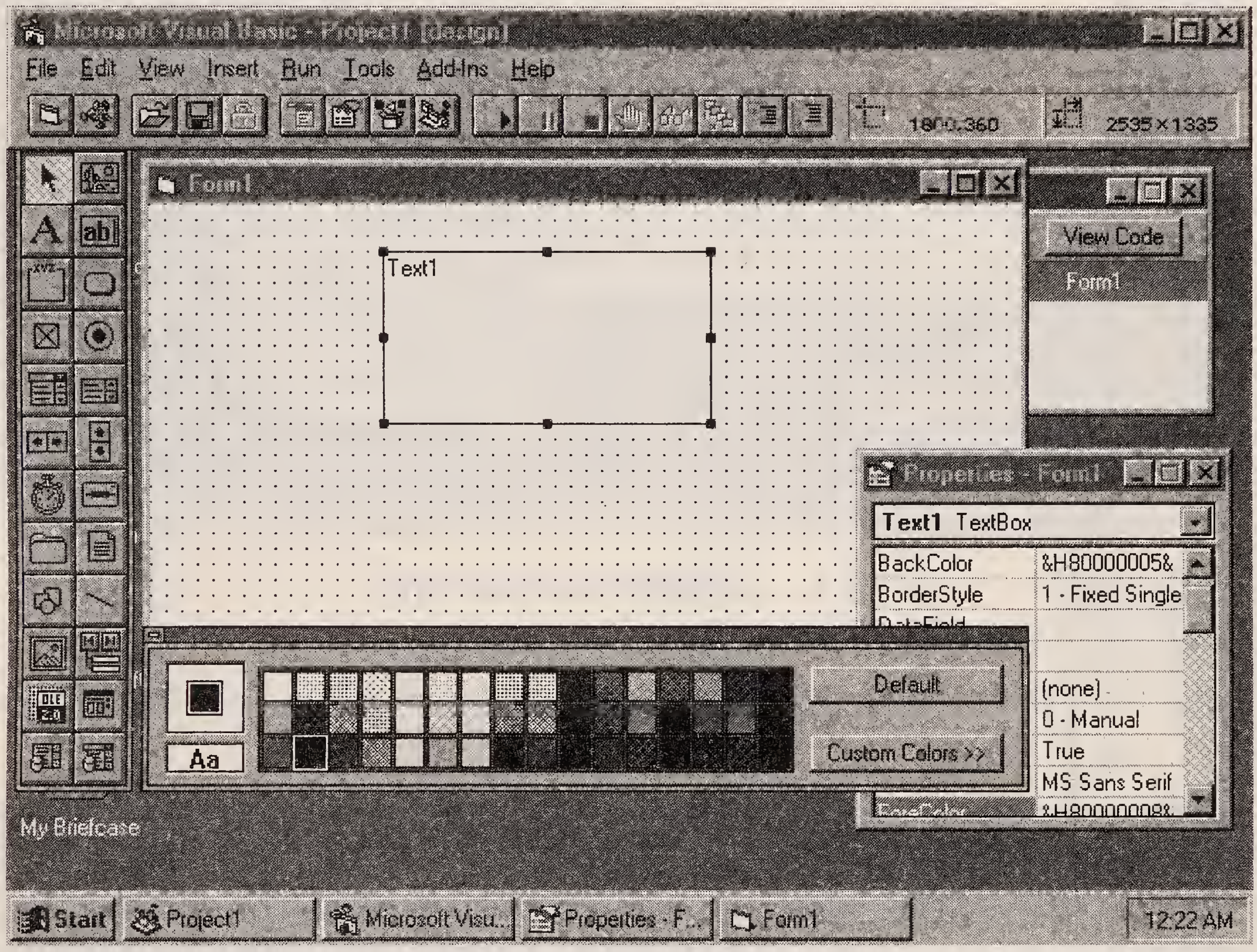
图 3.6 在 Form 窗口上连接对象

当我们想移动对象时，只需将鼠标移到该对象中，单击并拖动鼠标，移到自己满意的位置即可。

若想改变框架的形状，只需将鼠标移到包围该对象的周围的黑色小框架上，单击并拖动鼠标，即可改变框架的形状。

(5) Color Palette（调色板）窗口

调色板窗口一般是不可视的，我们可以通过 View 菜单中的 Color palette 选项来激活它，如图 3.7 所示。


Microsoft Visual Basic - Project1 [design]
File
Edit
View
Insert
Run
Tools
Add-Ins
Help
1800,360
2535×1335
Form1
Text1
View Code
Form1
Properties - Form1
Default
Custom Colors >>
Aa
My Briefcase
Start
Project1
Microsoft Visu...
Properties - F...
Form1
12:22 AM

Text1 TextBox

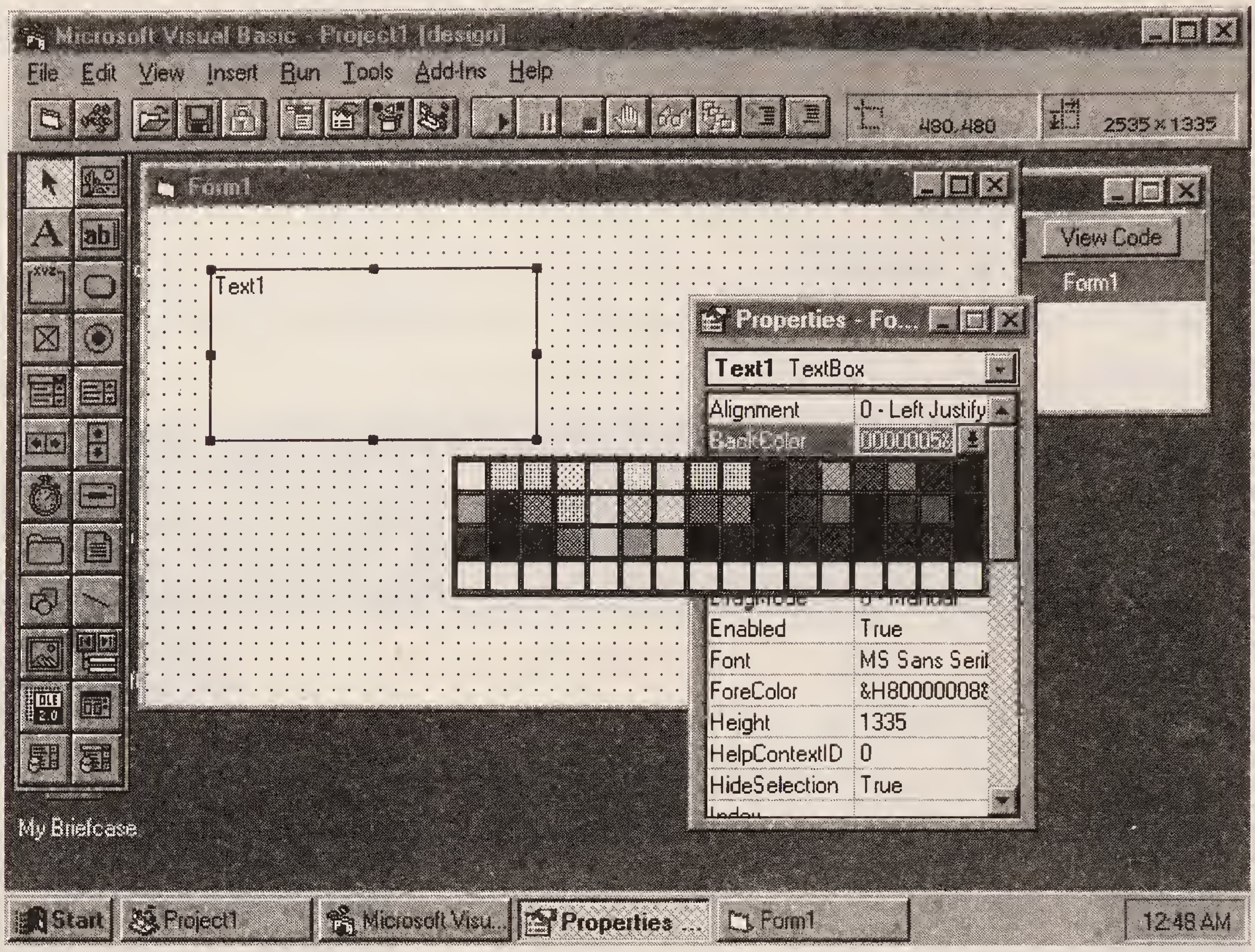
BackColor &H80000005&
BorderStyle 1 - Fixed Single
(none)
0 - Manual
True
MS Sans Serif

图 3.7　调色板窗口

Color Palette 窗口在图 3.7 的下方，我们看到窗口的左面有两个方框，上面的是一大方框套一小方框，下面的方框中有字母 Aa。上面的正方形方框就是给对象着色用的。当我们用鼠标单击内部的小方框时，下面将要选择的颜色就成为当前前景色。当单击外部的大方框时，下面将要选择的颜色就成为选中对象的背景色，并且所选的颜色当即在对象中显现出来。下面的那个长方形方框是用来显示结果用的，字母 Aa 的颜色表示当前前景色，后面的背景的颜色代表背景色，并随着用户选择的不同而变化。窗口的中间是一组可供用户选择的颜色，如果用户不喜欢这些颜色。可以单击窗口右边 Custom Colors 来自己定义颜色。也可以单击 Default（缺省值）来确认缺省值。

如果想关闭这个窗口，只需单击窗口左上角的小框，选择 Close（关闭）即可。

改变对象的颜色还可以通过 Properties（属性）窗口来实现。在窗口列表里选择 Forecolor 或 Backcolor，单击鼠标，即出现一向下的箭头，单击这一箭头，即会出现一组可供用户选择的颜色。选中什么颜色，单击一下这一颜色框即可。如图 3.8 所示。


Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
480,480
2535 x 1335
Form1
Text1
View Code
Form1
Properties - Fo...
My Briefcase
Start
Project1
Microsoft Visu...
Properties ...
Form1
12:48 AM

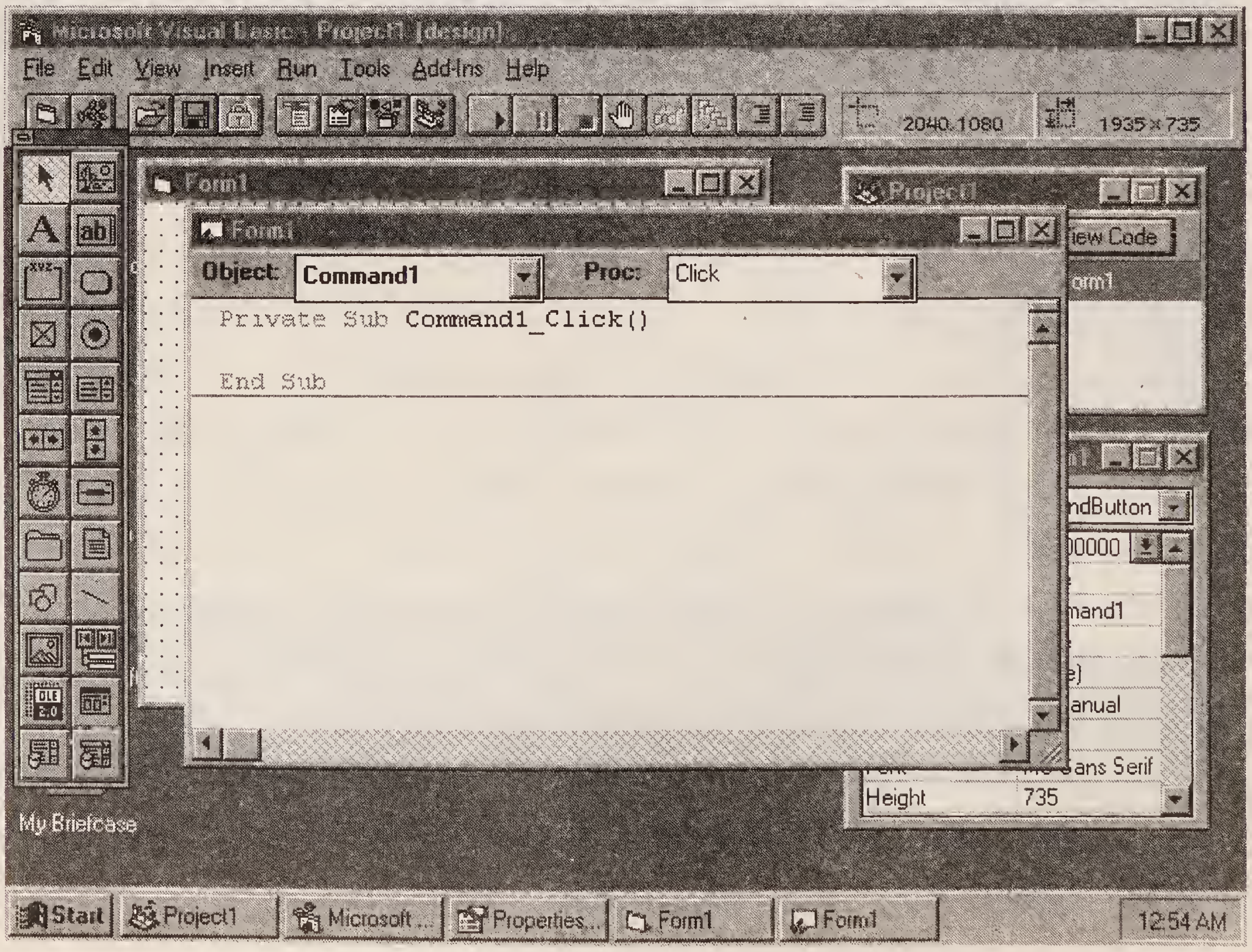
Text1 TextBox

Alignment	0 - Left Justify
BackColor	0000005
Enabled	True
Font	MS Sans Serif
ForeColor	&H80000008
Height	1335
HelpContextID	0
HideSelection	True

图 3.8　背景颜色(Backcolor)窗口

(6) Code(代码)窗口

代码窗口与所有 Form 窗口都有联系，且含有程序中所有 Form 和对象的代码，是进行编辑的窗口，如图 3.9 所示。


Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
2040,1080
1935 x 735
Form1
Project1
Form1
iew Code
orm1
ndButton
00000
mand1
anual
ans Serif
Height
735
My Briefcase
Start
Project1
Microsoft ...
Properties...
Form1
Form1
12:54 AM

Object: Command1　Proc: Click

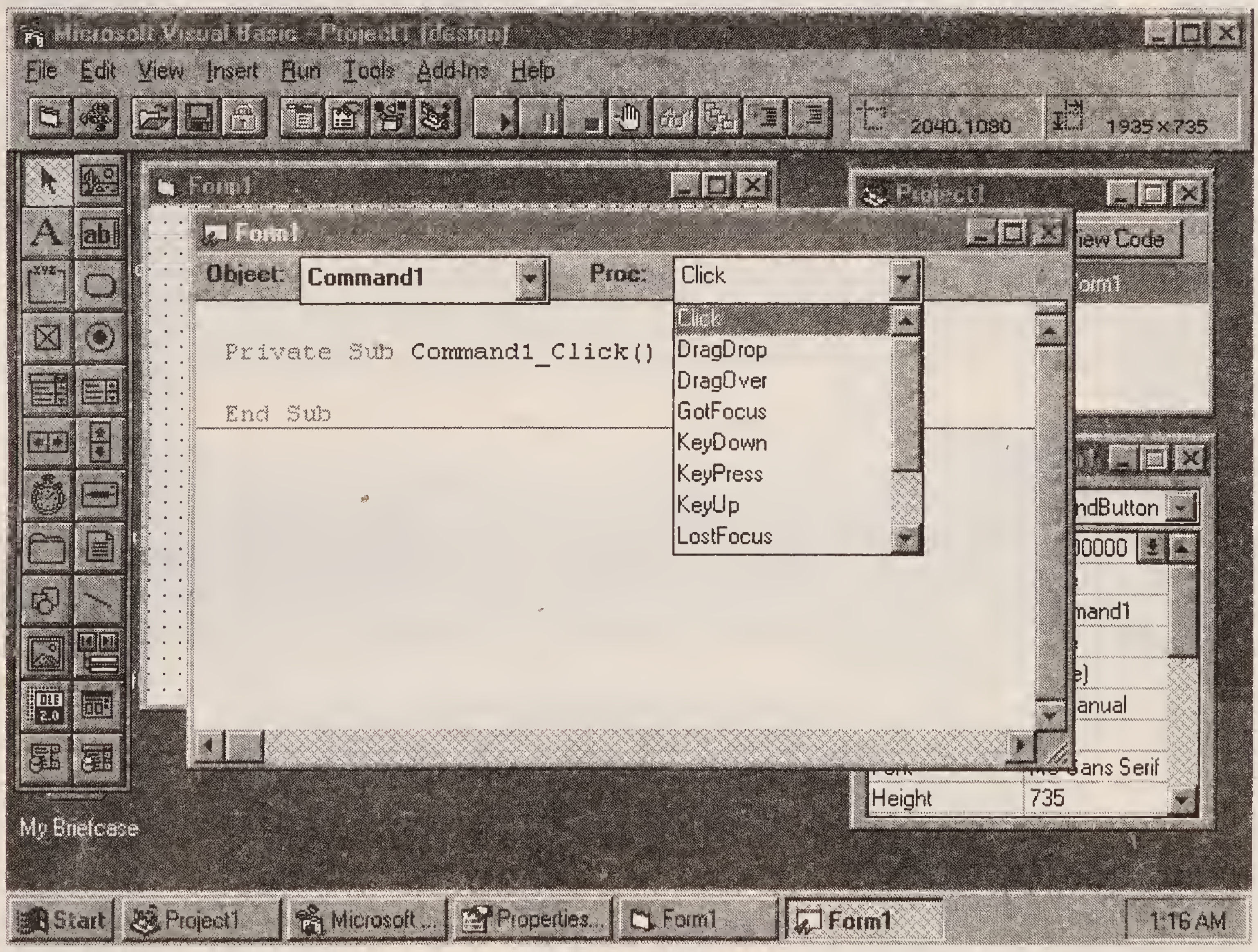
```
Private Sub Command1_Click()

End Sub
```

图 3.9　Code(代码)窗口

Code 窗口一般也是隐形的，我们可以通过选择 View 菜单中的 Code（代码）命令进入它。也可以通过单击 Project（工程）窗口中的 View code（打开代码窗口）来进入。还可以在 Form（窗体）窗口中双击鼠标左键来进入。

Code 窗口如图 3.9 所示，窗口的最上面一行为标题条。下面有两个下拉式菜单。左面的 Object（对象）包含有所有与 Form 相联系的对象，可以通过它右边的按钮把它们列出来。右边的 proc 即 procedure（过程）则包含有与当前选中对象相连接的所有事件，也可以通过它右边的按钮列出来。如图 3.10 所示。当用户选定一个 Object 和 Procedure 后，Visual Basic 则自动把过程头及过程尾列在窗口内，用户只要键入中间过程代码即可。

大家要注意，Visual Basic 4.0 的程序过程头加了一个 Private，而 Visual Basic 3.0 则没有。


Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
2040,1080
1935×735
Form1
Form1
Object: Command1
Proc: Click
Click
DragDrop
DragOver
GotFocus
KeyDown
KeyPress
KeyUp
LostFocus
Project1
iew Code
orm1
ndButton
00000
mand1
e)
anual
ans Serif
Height
735
My Briefcase
Start
Project1
Microsoft ...
Properties...
Form1
Form1
1:16 AM

```
Private Sub Command1_Click()

End Sub
```

图 3.10 Code（代码）窗口中的 Procedure 菜单

(7) Debug（调试）窗口

当一个应用程序在运行时遇到 STOP 语句，或执行暂停命令后就会弹出 Debug 窗口，如图 3.11 所示。我们也可以通过 View 菜单中的 Debug Window（调试窗口）命令来调出 Debug 窗口。

Debug 窗口中由两个面板组成，一个面板是 Watch Pane（观察面板），可以通过单击观察按钮来打开，在这个面板里可以显示正在观察的变量的当前值。

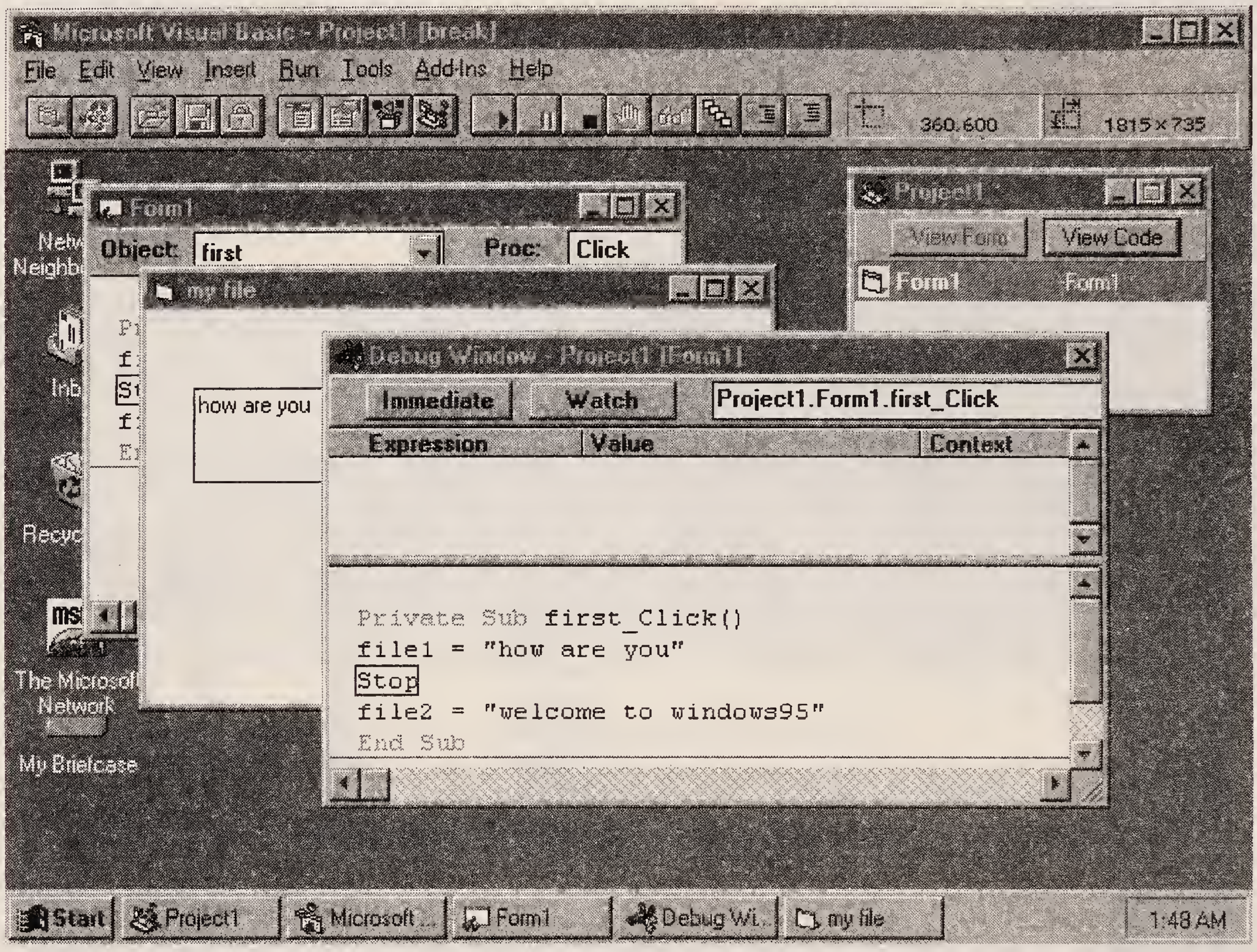
另一个面板则是 Immediate Pane（立即面板），当打开 Debug 窗口时，在没有设置的情况下首先看到的就是它。如果想在 Watch Pane（观察面板）显示时来打开它，单击上面的 Immediate 按钮即可。在暂停模式下，可以直接在这个面板键入代码并立即执行它。

在 Immediate 面板中虽不能保存代码，但可以把它们拷贝、粘贴到代码窗口中。

在暂停模式下，立即面板中的代码是在显示器上执行的。我们可以在过程（procedure）窗

口中观察到执行过程。例如，在立即窗口中键入 Print（变量名），则输出变量的值。就好像 Print 真的在过程中被执行一样，而其实程序是处于暂停状态的。

在这里，我们需要注意的是在图 3.11 中，Debug 窗口是由上下两部分组成的。上面那一部分即是 Debug 的立即以及观察面板，它们不是同时出现的。要想打开哪个面板，单击上面的那个按钮即可。下面的那部分则显示程序代码。


Microsoft Visual Basic - Project1 [break]
File Edit View Insert Run Tools Add-Ins Help
360,600
1815×735
Form1
Object: first
Proc: Click
my file
how are you
Debug Window - Project1 [Form1]
Immediate
Watch
Project1.Form1.first_Click
Expression
Value
Context
Project1
View Form
View Code
Form1
Network Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Start
Project1
Microsoft ...
Form1
Debug Wi...
my file
1:48 AM

```
Private Sub first_Click()
file1 = "how are you"
Stop
file2 = "welcome to windows95"
End Sub
```

图 3.11 Debug（调试）窗口

（8）Menu Editor（菜单编辑）窗口

Menu Editor 窗口也是隐藏的，我们可以通过单击 Tools（工具）菜单中的 Menu Editor 来打开它，如图 3.12 所示。

用户可以利用 Menu Editor 窗口来增加或修改一个窗体上的菜单，可以建立不同的菜单或往菜单里添加选项。可以改变选项的次序，建立级联菜单以及为菜单选项设置不同的属性。还可以将一个或多个菜单连到屏幕前台的 Form 窗口中。我们在后面的章节中将会详细讨论这一专题。

（9）Object Browser（对象浏览器）

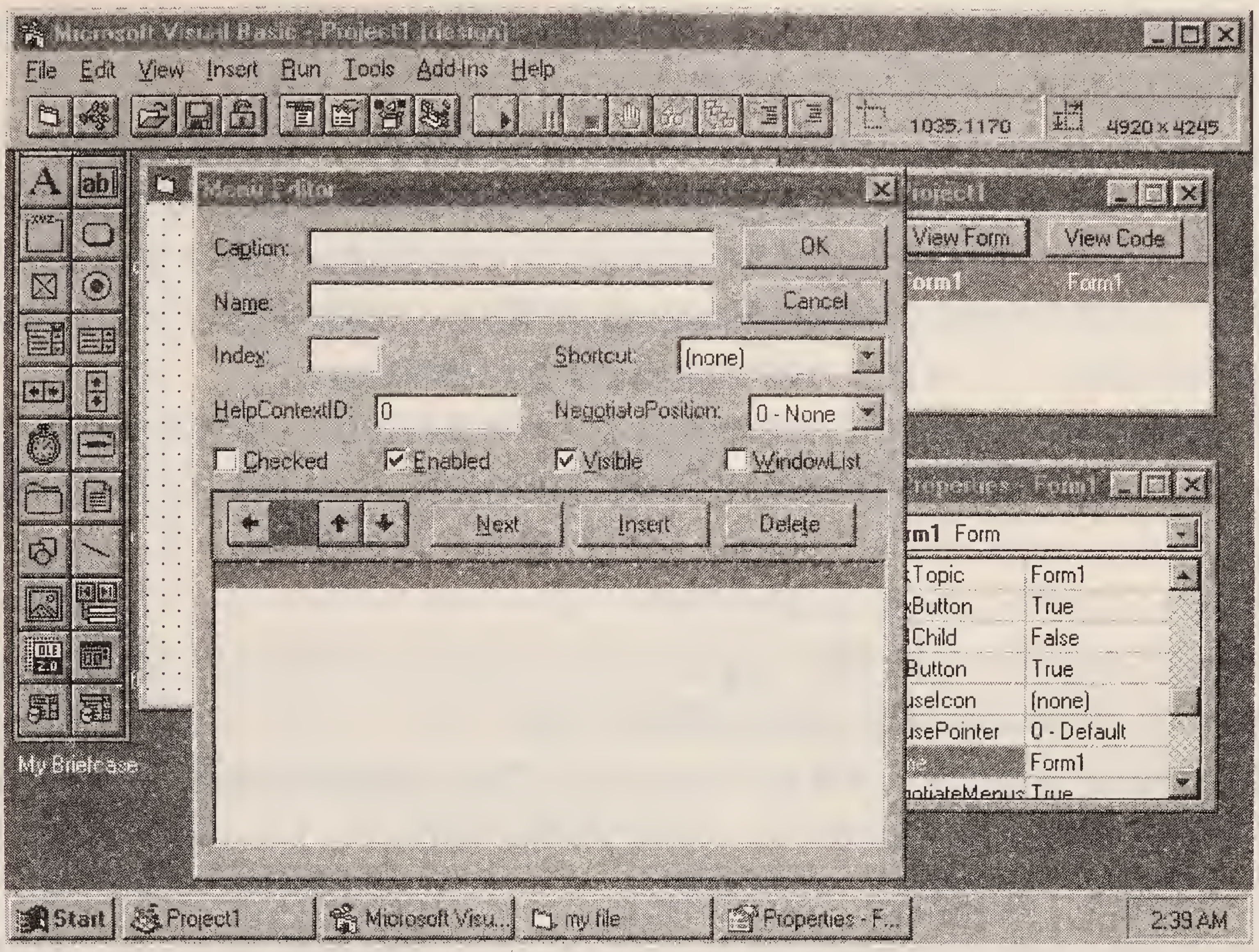
对象浏览器如图 3.13 所示，将在工程文件中显示出对象库中的类、属性和方法，并且可以显示模块以及过程。用户可以通过它来寻找和利用自己创建的对象。

要进入 Object Browser 有三种方法：

1）从 View 菜单中单击 Object Browser（对象浏览器）命令。

2）按 F2 键。

3）单击 Toolbar（工具栏）中的 Object Browser（对象浏览器）按钮。


Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
1035,1170
4920 x 4245
Menu Editor
OK
Cancel
Next
Insert
Delete
Project1
View Form
View Code
Properties - Form1
My Briefcase
Start
Project1
Microsoft Visu...
my file
Properties - F...
2:39 AM

Caption:

Name:

Index:

Shortcut: (none)

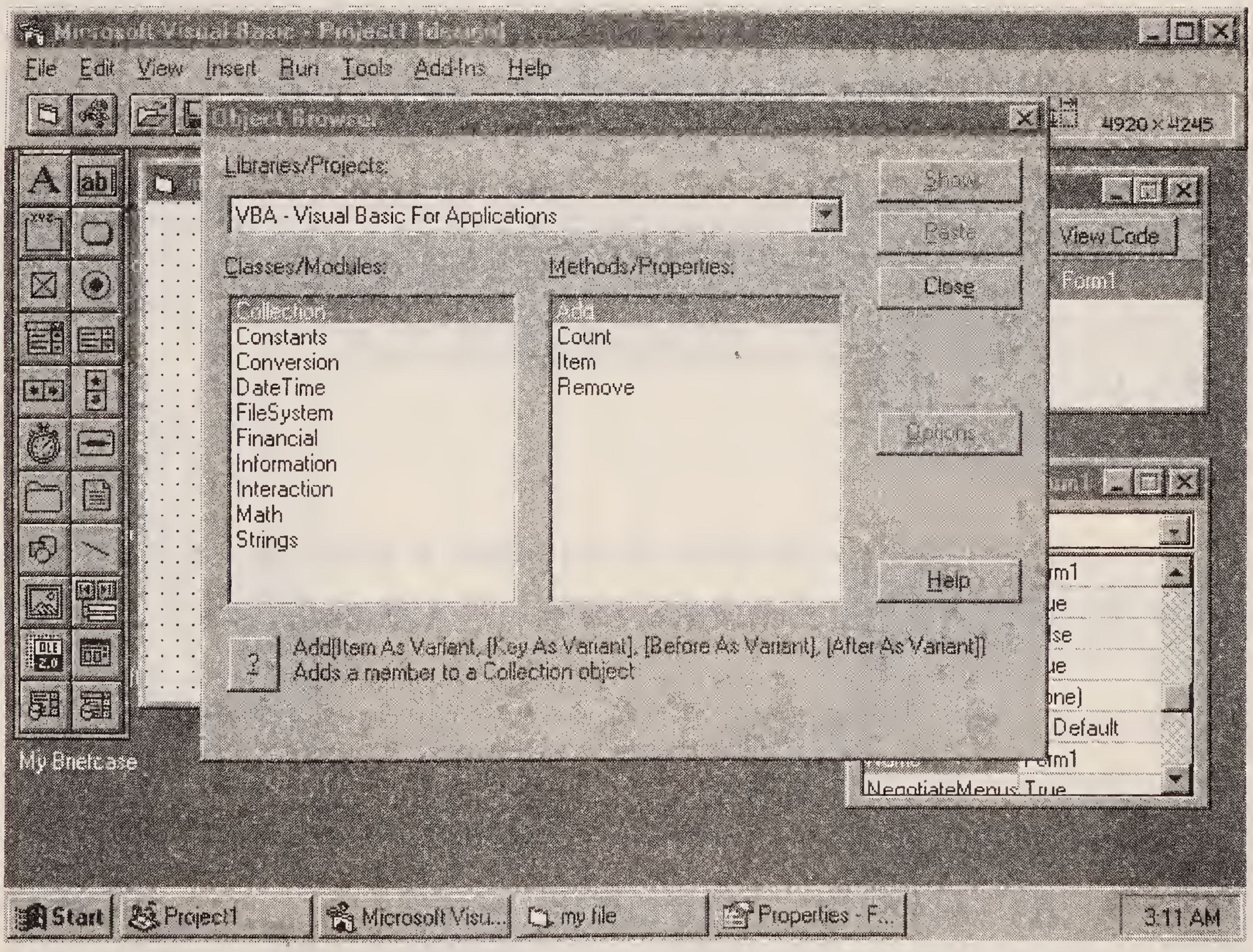
HelpContextID: 0

NegotiatePosition: 0 - None

☐ Checked ☑ Enabled ☑ Visible ☐ WindowList

Form1	Form1
rm1 Form	
Topic	Form1
Button	True
Child	False
Button	True
uselcon	(none)
usePointer	0 - Default
e	Form1
otiateMenus	True

图 3.12 Menu Editor(菜单编辑)窗口


Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
4920 x 4245
Object Browser
Show
Paste
Close
Options
Help
View Code
Form1
My Briefcase
Start
Project1
Microsoft Visu...
my file
Properties - F...
3:11 AM

Libraries/Projects: VBA - Visual Basic For Applications

Classes/Modules:	Methods/Properties:
Collection	Add
Constants	Count
Conversion	Item
DateTime	Remove
FileSystem	
Financial	
Information	
Interaction	
Math	
Strings	

Add(Item As Variant, [Key As Variant], [Before As Variant], [After As Variant])
Adds a member to a Collection object

图 3.13 Object Browser(对象浏览器)

要想关闭窗口，只要单击一下 Close（关闭）按钮，或单击窗口右上角的带×的按钮即可。

下面我们来介绍一下窗口中各对话框的功能：

Libraries/Projects（库/工程文件）对话框：显示与工程文件有关的对象库。用户可以选择相关的对象库，或自己的工程文件，来浏览一下它所包含的模块、过程、方法以及属性等等。

Classes/Modules（类/模块）：如果用户在库/工程文件对话框中选择了自己的工程文件，那么在这个对话框中将列出工程文件所包含的各种模块，以及用户所定义的类。如在库/工程文件对话框选择了一个对象库，这个框架将显示库中的所有对象。

通过选取类/模块对话框中的常数项可以查看一下 Visual Basic 中的固有常量。例如，在 Libraries/Projects 对话框中选择 Visual Basic 项，然后在 Classes/Modules 对话框中选择常数项，这样就可以在 Methods/Properties（方法/属性）对话框中选取一个常量（如 VbAlignLeft），然后把它粘贴到自己的代码中。

Methods/Properties（方法/属性）：如果已在 Libraries/Projects 对话框中选择了工程文件项，则在这个框架里显示已被定义的过程属性和方法。

“?”：在对象浏览器的左下角我们将会看到它。它显示的内容是，被选择的 Classes/Modules 或 Methods/Properties 对话框中某一项的在线帮助主题。

“Show”（显示）：在代码窗口中显示过程。

“Paste”（粘贴）：把一个方法或属性以及 Visual Basic 代码模板粘贴到一个过程中去。

“Options”（选项）：显示一个选项对话框，用户可以在自己的工程文件中添加关于被定义的方法、属性、模块或过程等的帮助信息与内容。

Visual Basic 的窗口不止这些，现在介绍的只是我们经常用到的几种，关于其他的窗口，在用到时，再给大家介绍。

在这里需要注意，几乎每个窗口的右上角都有三个小按钮，其中在最右边的带×的按钮表示关闭窗口，中间带口的按钮表示放大/复原窗口，左边的带_的按钮表示将窗口缩为一图标。一般窗口的左上角也都有一图标，单击这一图标会弹出一菜单，让用户选择关闭、缩小和放大等命令。

3.2.2 工具栏

工具栏中的按钮所代表的命令都是我们经常用到的。要想执行某一命令，只要单击一下这个按钮即可。在工具栏的右边有两个文本框，分别显示当前选中的窗体或控件的位置与尺寸。

Form（窗体）——建立一个新窗体。
Module（模块）——建立一个新模块。
Open Project（打开工程文件）——打开一个工程文件。
Save Project（保存工程文件）——保存当前的工程文件。
Lock Controls（锁定控件）——锁定或打开活动窗体中的控件。
Menu Editor（菜单编辑）——打开菜单编辑窗口。
Properties（属性）——打开属性窗口。
Object Browser（对象浏览器）——打开对象浏览器。

Project(工程文件)——打开工程文件窗口。

Start(开始)——运行一个应用程序。

Break(暂停)——在运行程序时暂停。

End(结束)——结束一个应用程序的运行并回到设计窗口。

Toggle Breakpoint(设置断点)——设置或清除当前行的断点。

Instant Watch(立即观察)——在代码窗口中显示选中变量的值。

Calls(调用)——打开 Calls(调用)窗口。

Step Into(单进入)——跟踪每段代码并进入过程。

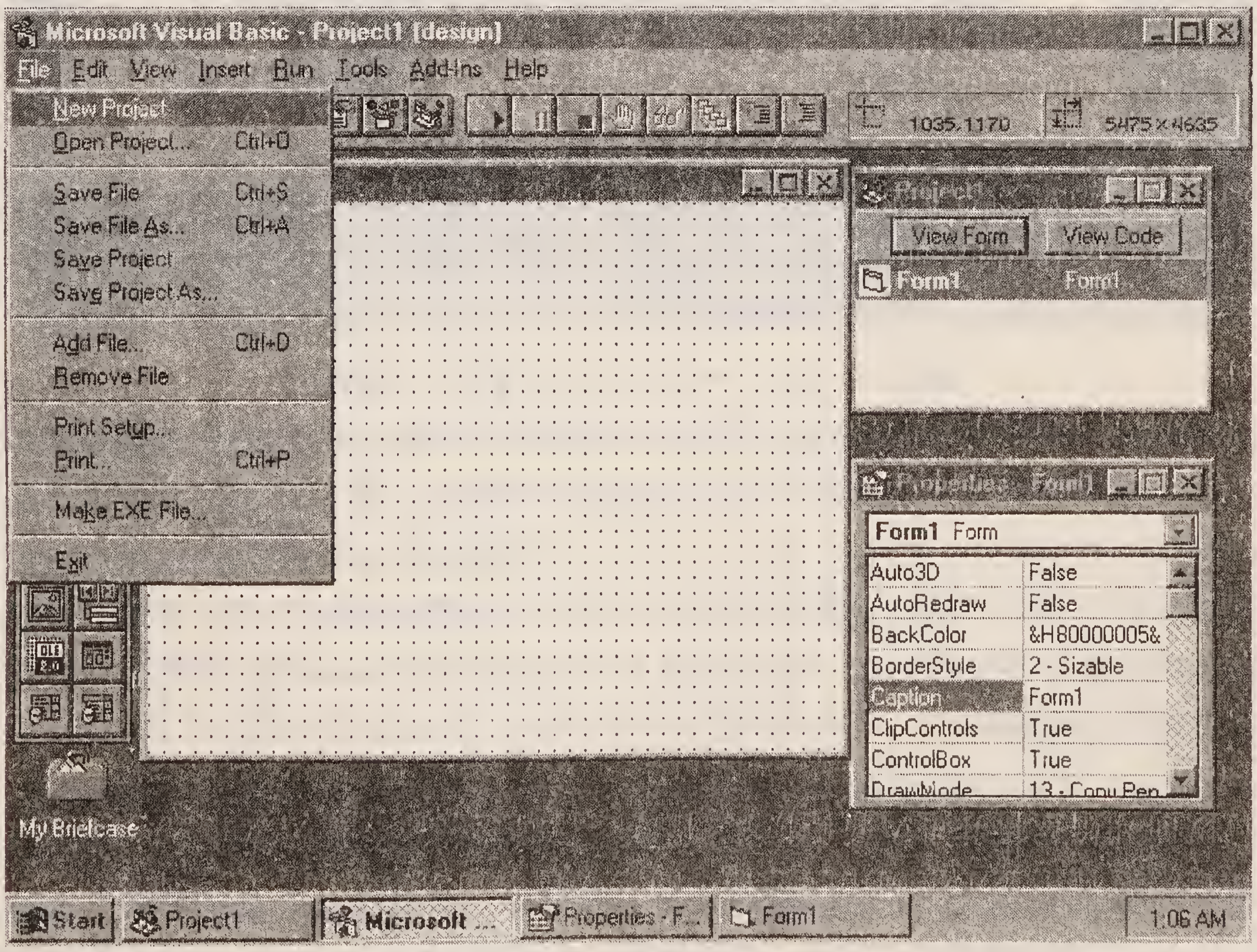
Step Over(单步执行)——在代码窗口中每次只执行一个过程或一个语句。

3.2.3　菜单

在 Visual Basic 4.0 中有八个菜单，它们分别是 File(文件)，Edit(编辑)，View(查看)，Insert(插入)，Run(运行)，Tools(工具)，Add-Ins(附件)，Help(帮助)。它与 Visual Basic 3.0 的菜单有所不同，它没有 Debug(调试)，Options(选项)，Window(窗口)菜单，而增加了 Insert(嵌入)，Tools(工具)，Add-Ins(附件)菜单。

(1)　File 菜单(文件菜单)

用鼠标在菜单条的 File(文件)上单击，即弹出一菜单，如图 3.14 所示。File 菜单共有 12 条命令，它们分别是 New Project(新工程文件)，Open Project(打开工程文件)，Save File(保存文件)，Save File As(另存文件为)，Save Project(保存工程文件)，Save Project As(另存工程文件为)，Add File(添加文件)，Remove File(删除文件)，Print Setup(打印 Setup 文件)，Print(打印)，Make EXE 文件(建立 EXE 文件)和 Exit(退出)。


Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
New Project
Open Project... Ctrl+O
Save File Ctrl+S
Save File As... Ctrl+A
Save Project
Save Project As...
Add File... Ctrl+D
Remove File
Print Setup...
Print... Ctrl+P
Make EXE File...
Exit
1035,1170
5475 x 4635
Project
View Form
View Code
Form1 Form1
Properties - Form1
My Briefcase
Start
Project1
Microsoft ...
Properties - F...
Form1
1:06 AM

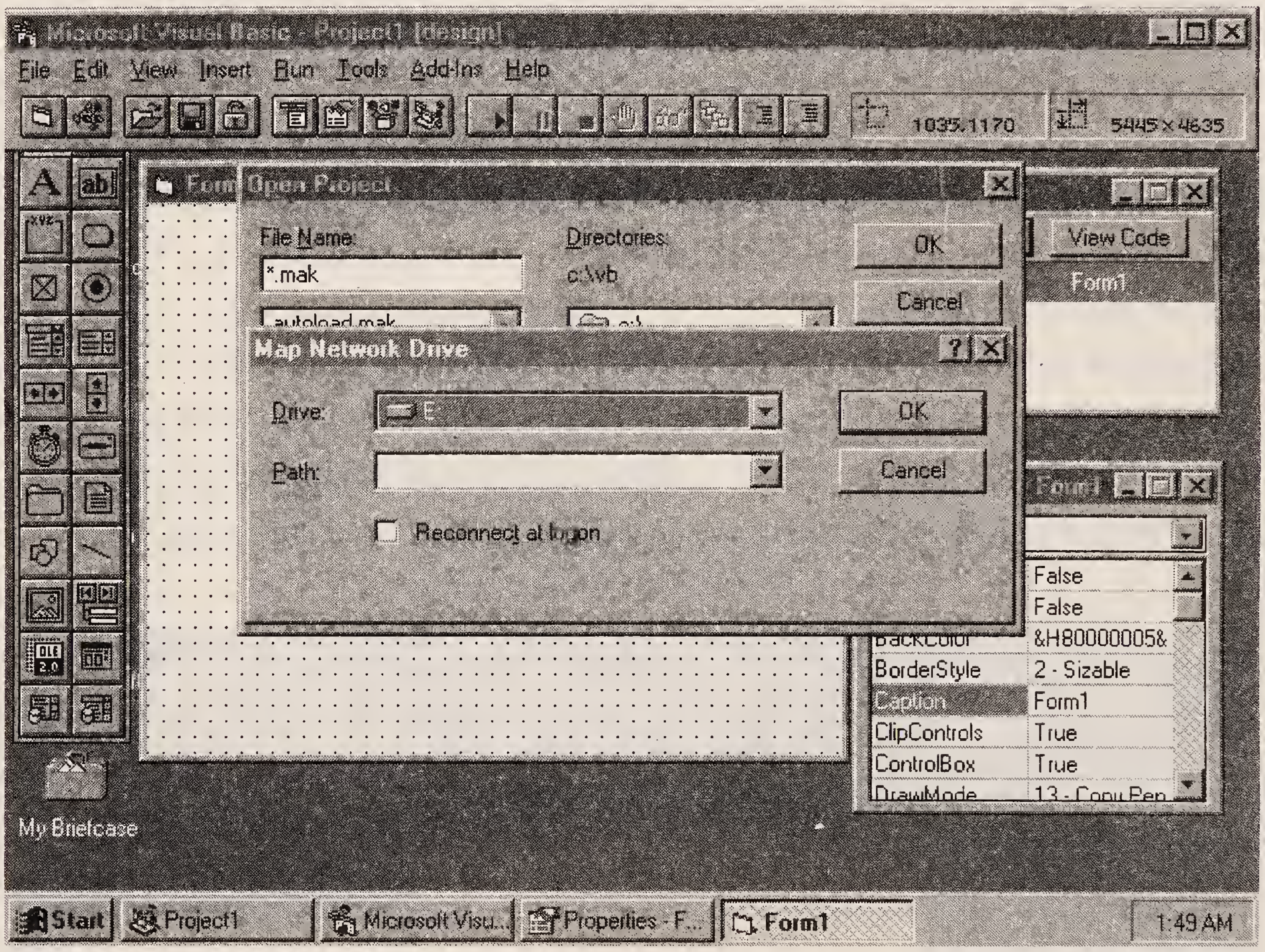
Form1 Form

Auto3D	False
AutoRedraw	False
BackColor	&H80000005&
BorderStyle	2 - Sizable
Caption	Form1
ClipControls	True
ControlBox	True
DrawMode	13 - Copy Pen

图 3.14　File(文件)菜单

New Project 命令用于建立一个新工程文件。用户一次只能打开一个工程文件。当建立一个新工程文件时，Visual Basic 会提醒用户保存当前的工程文件，然后所有当前工程文件中的文件将被关闭，并且建立一个新的文件。用户可以通过改变 AUTOLOAD.MAK 文件中的窗体模块、类模块、标准模板、定制模块、环境选项和格式选项等来为新建立的工程文件设定缺省值。

Open Project 命令用于关闭当前的工程文件，装入一个新的工程文件，快捷键是 Ctrl+D，即 Ctrl 与 D 键同时按下。也可单击 Toolbar（工具栏）中的 Open Project 按钮。Visual Basic 执行这一命令时会出现一对话框，用户只要输入要打开的工程文件名即可。如果用户微机已联网，想要查看其他用户中的工程文件，只要选择 Network 按钮，即出现一对话框，用户只要键入工程文件的路径及驱动器名即可，如图 3.15 所示。


File Name: *.mak

Directories: c:\vb

Drive: E

	False
	False
BackColor	&H80000005&
BorderStyle	2 - Sizable
Caption	Form1
ClipControls	True
ControlBox	True
DrawMode	13 - Copy Pen

Microsoft Visual Basic - Project1 [design]
File
Edit
View
Insert
Run
Tools
Add-Ins
Help
1035,1170
5445 x 4635
Form
Open Project
OK
Cancel
autoload.mak
c:\
View Code
Form1
Map Network Drive
Path:
OK
Cancel
Reconnect at logon
Form1
My Briefcase
Start
Project1
Microsoft Visu...
Properties - F...
Form1
1:49 AM

图 3.15　Open Project（打开工程文件）对话框

Save File 和 Save File As 命令用于把用户命名的文件保存在磁盘上。快捷键是 Ctrl+S，Ctrl+A。如果是第一次存盘，Save File 命令将弹出一对话框，用户只要把文件名及路径输入即可。如果已存在这样一个文件名，则 Visual Basic 将提醒用户是否覆盖它。用户也可以用缺省名作为其扩展名，即.FRM 为窗口扩展名，.CLS 为类模块扩展名，.BAS 为标准模块扩展名。如果用户想把文件保存在某一联网微机上，只要选择 Network 键，在对话框中键入路径及驱动器名和文件名即可。

Save Project 与 Save Project As 命令用于把工程文件保存在磁盘上，快捷键是 Toolbar（工具栏）中的 Save Project 按钮。当 Visual Basic 执行这一命令时，会弹击一对话框，用户只要把工程文件的名字、路径、所在的磁盘以及工程文件中所包含的文件类型输入即可。如果是第一次存盘，当用户单击 Save Project 命令时，Visual Basic 先弹出 Save File As 对话框，输入文件名后，会又弹出 Save Project As 对话框。

Add File 命令用于把窗体模块、MDI(multiple-document interface 多文档界面)Form、类模块以及源文件等加到工程文件中。用户可以分享工程文件之间的文件。当文件被改动并被存盘时，用户的改动会影响所有包含此文件的工程文件，快捷键是 Ctrl+D。用户可以通过工具菜单中的定制控制对话框，向工程文件的工具框中添加定制控件及对象。Visual Basic 执行这一命令后会弹出一对话框，键入文件名及路径即可。

Remove File 命令用于删除工程文件中的窗体模块、类模块和源文件等。如果在上次存盘时文件有所改动，Visual Basic 将会提醒用户是否保留它。在工程文件中被删除的文件实际上还存在于硬盘中，直到用户把它删掉为止。用户可以通过定制控制对话框从工具框中删掉定制控件。但是如果这个控件还在一个窗体文件中，就无法从工具框中将它删除。

Print Setup 命令用于显示一个标准的打印设置对话框，用户可以根据打印机的具体情况进行设置。

Print 命令用于显示一对话框，由用户来决定打印范围、内容、质量等内容。快捷键是 Ctrl+P。

Make .EXE File 命令用于建立一个可执行的应用程序文件。这样一个文件不需要 Visual Basic 就可执行。但运行时 VBRUN400. DLL 必须存在于\WINDOWS\SYSTEM 或在用户设置的路径中。对于不同的可执行文件，它们需要不同的. DLL 文件。这个命令不会影响到工程文件，用户仍旧可以对源程序进行修改，而对可执行文件不会产生任何影响。在执行这个命令时会弹出一对话框，用户只需把路径及文件名输入即可。单击 Options(选项)按钮后又会弹出一对话框，可以为应用程序设置名称及图标。

(2) Edit(编辑)菜单

Edit 菜单如图 3.16 所示，共有 15 条命令，分别是 Undo(取消)、Redo(重新执行)、Cut(剪切)、Copy(复制)、Paste(粘贴)、Paste Link(粘贴连接)、Clear(清除)、Find(寻找)、Replace(替换)、Indent(缩排)、Outdent(扩排)、Bring To Front(带到前台)、Send To Back(送到后台)、Align To Grid(与网格对齐)、Lock Controls(锁定控件)。

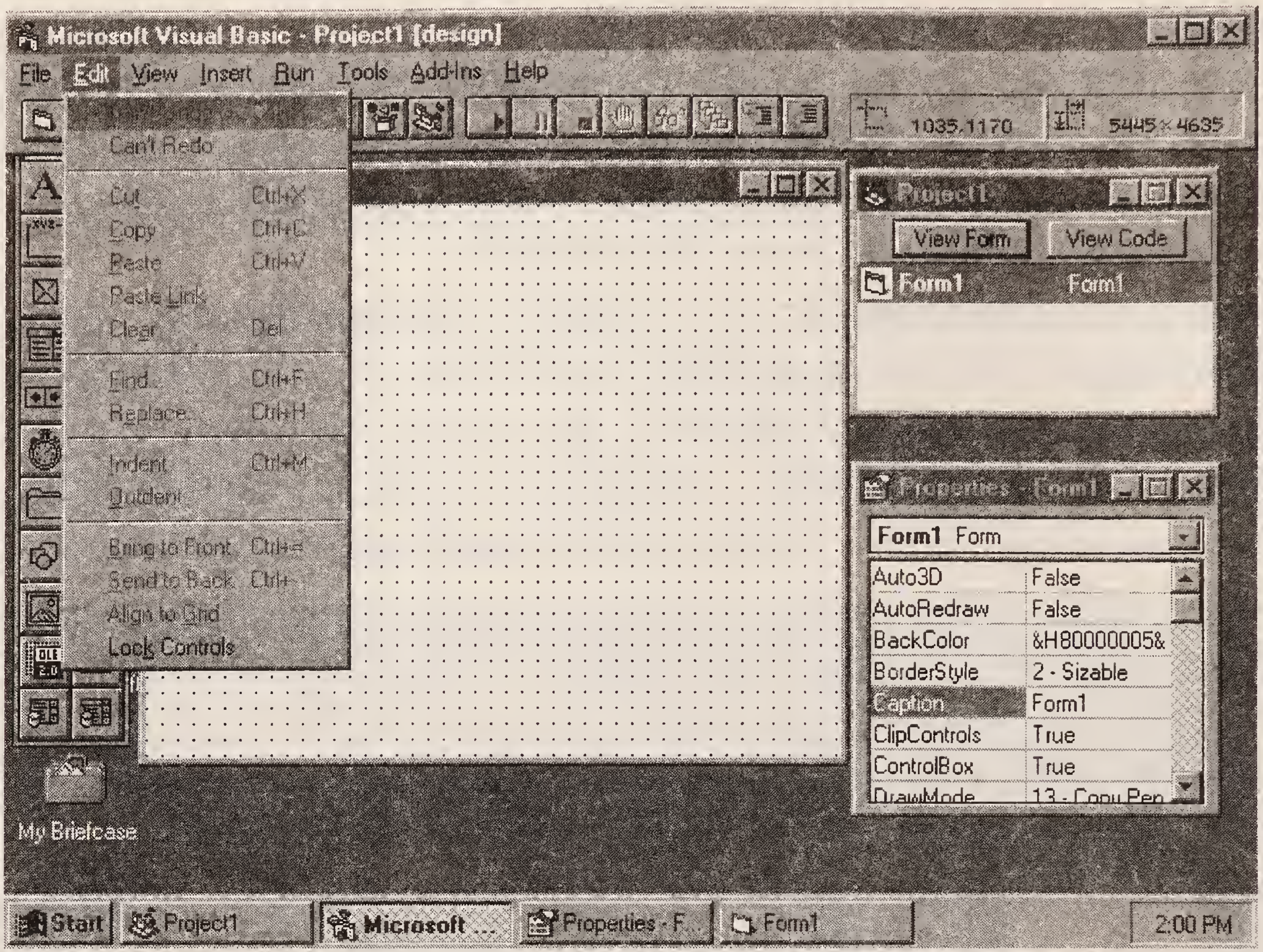
Undo 命令用于取消上次的编辑命令，例如删除控件等。当用户删除一个或多个控件以后，可以通过恢复命令来恢复被删除的控件和它们的属性。但 Tabindex 属性不能被恢复。对于窗体文件，如果在执行了控件删除命令后又被编辑过，则 Undo 命令不起作用。快捷键是 Ctrl+Z。

在 Undo 命令被执行后没有进行任何动作，而 Redo 命令将恢复被 Undo 之前的文本编辑内容。快捷键是 Ctrl+Backspace。如果在 Undo 之前没做过任何编辑或编辑后又执行了其他动作，则 Redo 命令是无效的。

Cut 命令用于把选中的控件或文本内容剪切到剪贴板上，在执行这一命令前要至少选定一个控件。快捷件是 Ctrl+X 或 Shift+DEL。

Copy 命令用于把选中的控件或文本内容复制到剪贴板上。快捷键是 Ctrl+C 或 Ctrl+Ins。

Paste 命令用于把剪贴板的内容转移到工程文件中。文本文件被放置在插入点，控件被放置在活动窗体的左上角。当粘贴一个控件时属性不变。用户可以保留原来的名字，并设置一个控件数组(control array)，也可以改用一个新名字。当把一个控件粘贴到另一个控件时，例如把一个控件粘贴到 Frame(框架)中时，要先选择好框架，然后把控件粘贴到里面。快捷


Microsoft Visual Basic - Project1 [design]
File
Edit
View
Insert
Run
Tools
Add-Ins
Help
1035,1170
5445 x 4635
Can't Redo
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Paste Link
Clear Del
Find... Ctrl+F
Replace... Ctrl+H
Indent Ctrl+M
Outdent
Bring to Front Ctrl+=
Send to Back Ctrl+-
Align to Grid
Lock Controls
Project1
View Form
View Code
Form1 Form1
Properties - Form1
Form1 Form
My Briefcase
Start
Project1
Microsoft ...
Properties - F...
Form1
2:00 PM

Auto3D	False
AutoRedraw	False
BackColor	&H80000005&
BorderStyle	2 - Sizable
Caption	Form1
ClipControls	True
ControlBox	True
DrawMode	13 - Copy Pen

图 3.16 Edit(编辑)菜单

键是 Ctrl+V 或 Shift+Ins。

Clear 命令用于删除当前选中的控件或文本。要想把一个窗体或模块从工程文件中删除,而仍旧保持在磁盘中,可以从 File(文件)菜单中选择 Remove File(删除文件)命令。想从磁盘中删除文件,则选择操作系统中的删除命令。快捷键是 DEL。

Paste Link 命令支持粘贴连接到一个 DDE(dynamic data exchange 动态数据交换)库。只有当剪贴板中有一个 DDE 库,且选中的控件是一个 DDE 宿(sink)时,这个命令才是合法的。用户还可以利用 OLE(对象的链接与嵌入)控件,用 Paste Link 命令连接其他应用程序中的数据。当数据被复制到剪贴板上后,选择窗体中的 OLE 控件,再用 Paste Link 命令来连接数据。

Find 命令用来寻找文本文件,快捷键是 Ctrl+F(寻找),F3(寻找下一个)、Shift+F3(寻找上一个)。如果寻找的文件存在,则对话框将被关闭,被选中的文件将显示出来。若找不到,则 Visual Basic 显示一个信息,告诉用户文件找不到。

Replace 命令在工程文件中选择一个文本文件的代码,并用一个新的文本文件来代替它,快捷键是 Ctrl+H。

Indent 命令用于把选中的一行代码整体向后移动四格。相当于 Tab 键的功能。

Outdent 命令用于把选中的一行代码整体向前移动四格,各代码之间的位置保持不变。

Brint To Front 命令用于把选中的对象移到最前台。快捷键是 Ctrl+=。

Send To Back 命令用于把选中的对象移到后台。快捷键是 Ctrl+-。在代码方式下,可以用 ZOrder 方法把窗体或控件移到前台或后台。

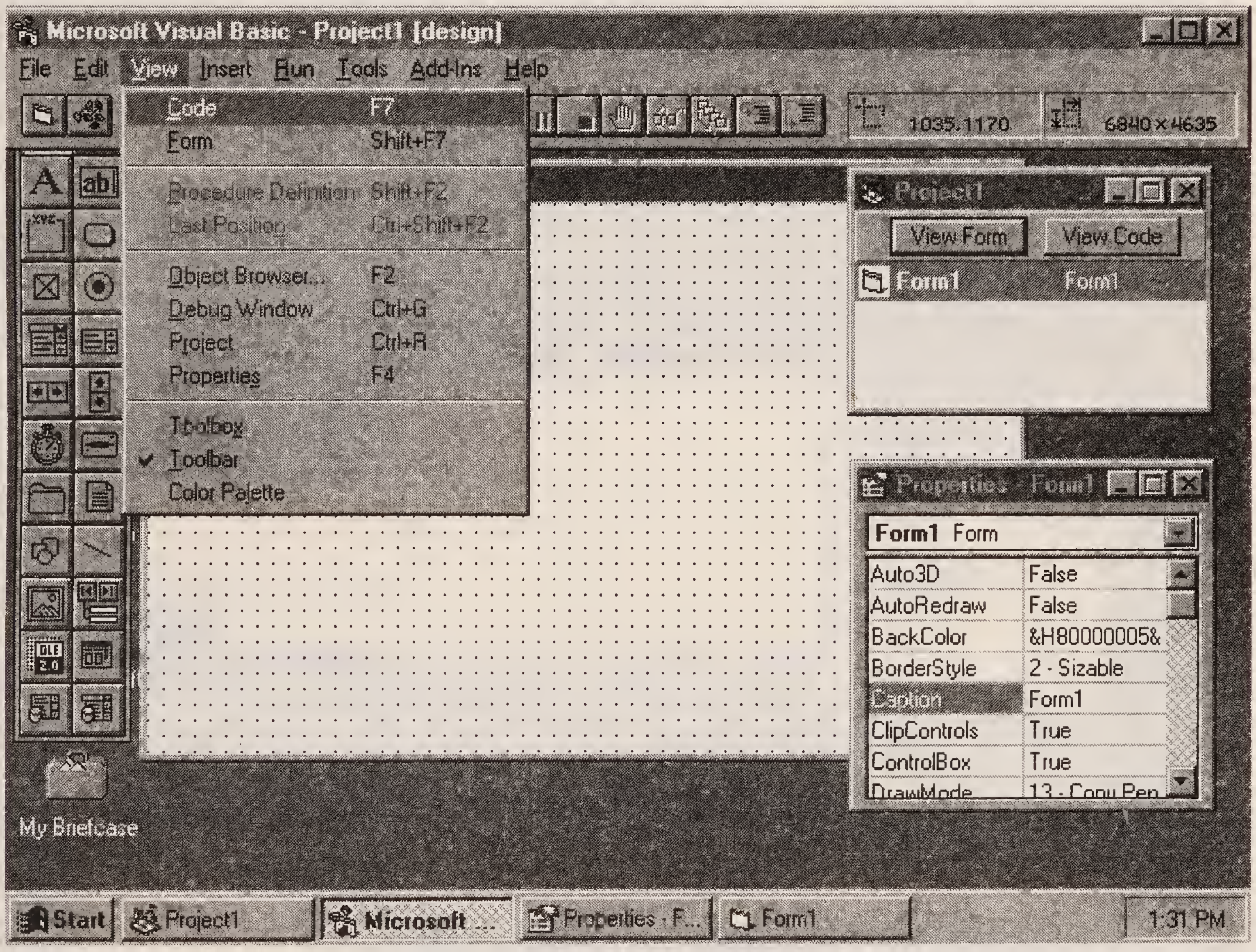
Align To Grid 命令使选中的 Form 窗口中的控件与网格自动对齐。要使所有的控件都与网格自动对齐,只要选择 Environment Options(环境选项)对话框中的 Align Control To

Grid(将控件与网格对齐)的命令即可。

Lock Controls 命令用于把控件和窗体锁定在它们的当前位置上,这样,就不会出现用户无意的移动它们的情况了。这一命令只能锁定某一窗体中的控件,在其他窗体中的控件是不会受到影响的,Visual Basic 将会自动跟踪哪些窗体中的控件被锁定而哪些没有。快捷键是工具栏中的 Lock Controls 命令。

(3) View(查看)菜单

View 菜单如图 3.17 所示,由 11 个命令组成,它们分别是 Code(代码)、Form(窗体)、Procedure Definition(过程定义)、Last Position(上次位置)、Object Browser(对象浏览器)、Debug Windows(调试窗口)、Project(工程文件)、Properties(属性)、Toolbox(工具框)、Toolbar(工具栏)、Color Palette(调色板)。


Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
Code F7
Form Shift+F7
Procedure Definition Shift+F2
Last Position Ctrl+Shift+F2
Object Browser... F2
Debug Window Ctrl+G
Project Ctrl+R
Properties F4
Toolbox
✓ Toolbar
Color Palette
1035,1170
6840×4635
Project1
View Form
View Code
Form1 Form1
Properties - Form1
Form1 Form
My Briefcase
Start
Project1
Microsoft ...
Properties - F...
Form1
1:31 PM

Auto3D	False
AutoRedraw	False
BackColor	&H80000005&
BorderStyle	2 - Sizable
Caption	Form1
ClipControls	True
ControlBox	True
DrawMode	13 - Copy Pen

图 3.17 View(观察)菜单

Code 命令用于显示一个活动的代码窗口,快捷键是 F7。

Form 命令用于显示一个 Form 窗口,快捷键是 Shift+F7。

Procedure Definition 命令用于为一个过程显示代码。例如在 Form _ Load 过程中,可以调用 mnuOpen _ Clink 过程,然后选择 Procedure Definition(过程定义)命令,Code 窗口即弹出与 mnuOpen _ Click 事件过程相关的代码。快捷键是 Shift+F2。

Last Position 命令允许用户快速浏览代码中以前的一个单元。只有在 Code(代码)窗口或 Debug(调试)窗口中编辑代码或执行 Procedure Definition(过程定义)命令时,这一命令才会有效。例如,在一个代码窗口中,改变第一行内容,编辑第二行,用过程定义命令跳到第三行,编辑第四行,然后转到第五行。当选择 Last Position(上一位置)命令时,首先显示第四行内容。如果再选择 Last Position 命令,第三行将被显示。Visual Basic 只跟踪被编辑过的前四行内容。快捷键是 Ctrl+Shift+F2。

Object Browser 命令用于打开对象浏览器窗口，快捷键是 F2，或使用工具栏中的 Object Browser 按钮。当程序运行时这一命令是无效的。

Debug Windows 命令用于打开一个调试窗口，快捷键是 Ctrl+G。

Project 命令显示一个工程文件窗口，快捷键是 Ctrl+R 或使用工具栏中的 Project 按钮。

Properties 命令用于打开一个属性窗口，快捷键是 F4 或使用工具栏中的 Properties 按钮。

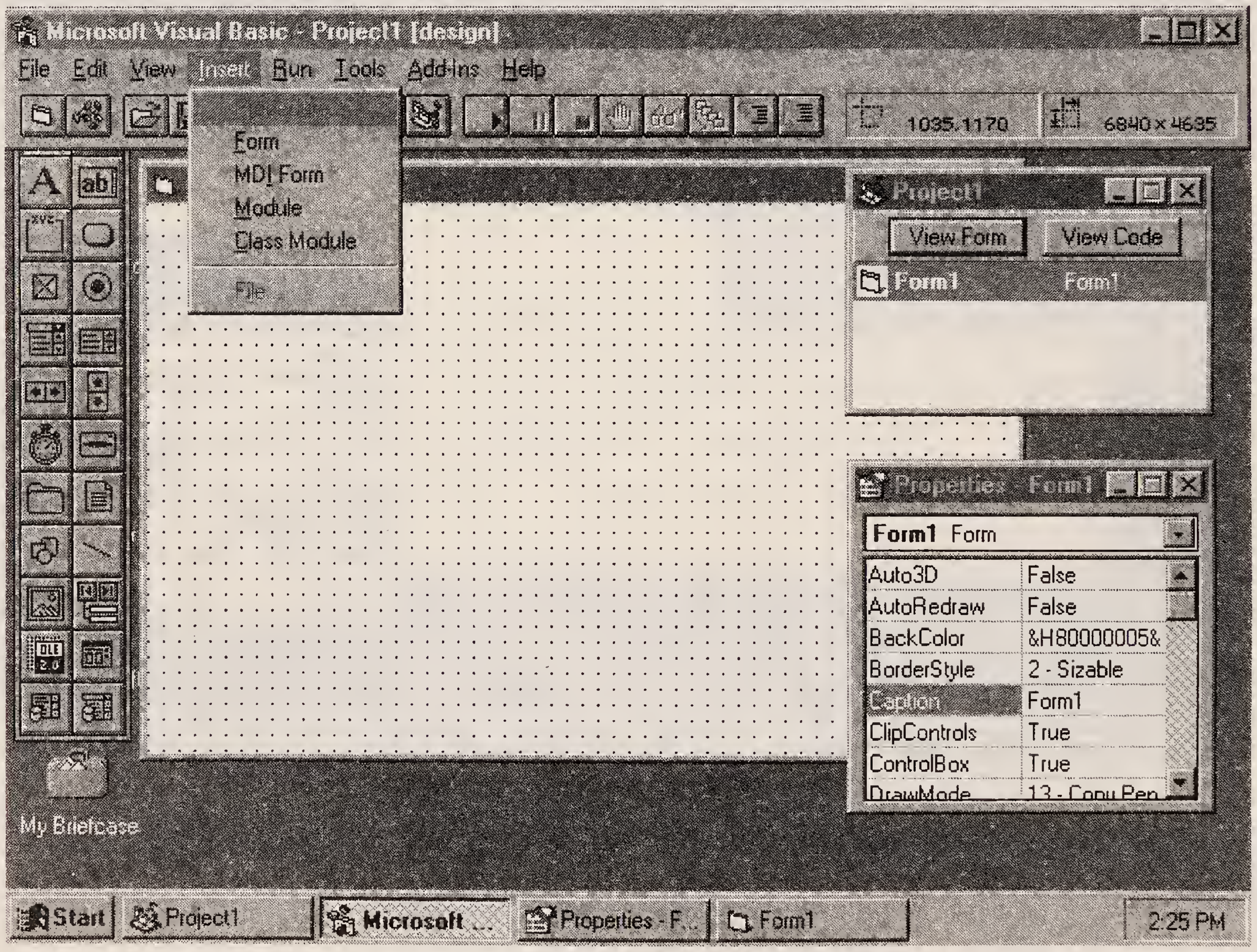
Toolbox 命令用于显示工具框。

Toolbar 命令用于显示或关闭工具栏。

Color Palette 命令用于显示调色板。

(4) Insert(插入)菜单

Insert 菜单如图 3.18 所示，它共有六条命令，分别是 Procedure(过程)、Form(窗体)、MDI Form(MDI 窗体)、Module(模块)、ClassModule(类模块)、File...(文件...)。


Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
Form
MDI Form
Module
Class Module
File
1035,1170
6840 x 4635
Project1
View Form
View Code
Form1 Form1
Properties - Form1
Form1 Form
My Briefcase
Start
Project1
Microsoft ...
Properties - F...
Form1
2:25 PM

Auto3D	False
AutoRedraw	False
BackColor	&H80000005&
BorderStyle	2 - Sizable
Caption	Form1
ClipControls	True
ControlBox	True
DrawMode	13 - Copy Pen

图 3.18 Insert(嵌入)菜单

Procedure 命令用于插入一个新的 Sub(子程序)、Function(功能)和 Property(属性)过程。Insert Procedure 对话框允许用户对于某过程可视或不可视进行设置。

Form 命令用于显示与当前选定的窗体模块相关的窗体窗口，快捷键是 Shift+F7。

File 命令用于把其他文件中的文本文件嵌入到某一模块中，把光标移动到想要嵌入文件的地方，单击 File 命令，然后在插入文件对话框中键入文件名即可，在程序运行时这一命令是无效的。

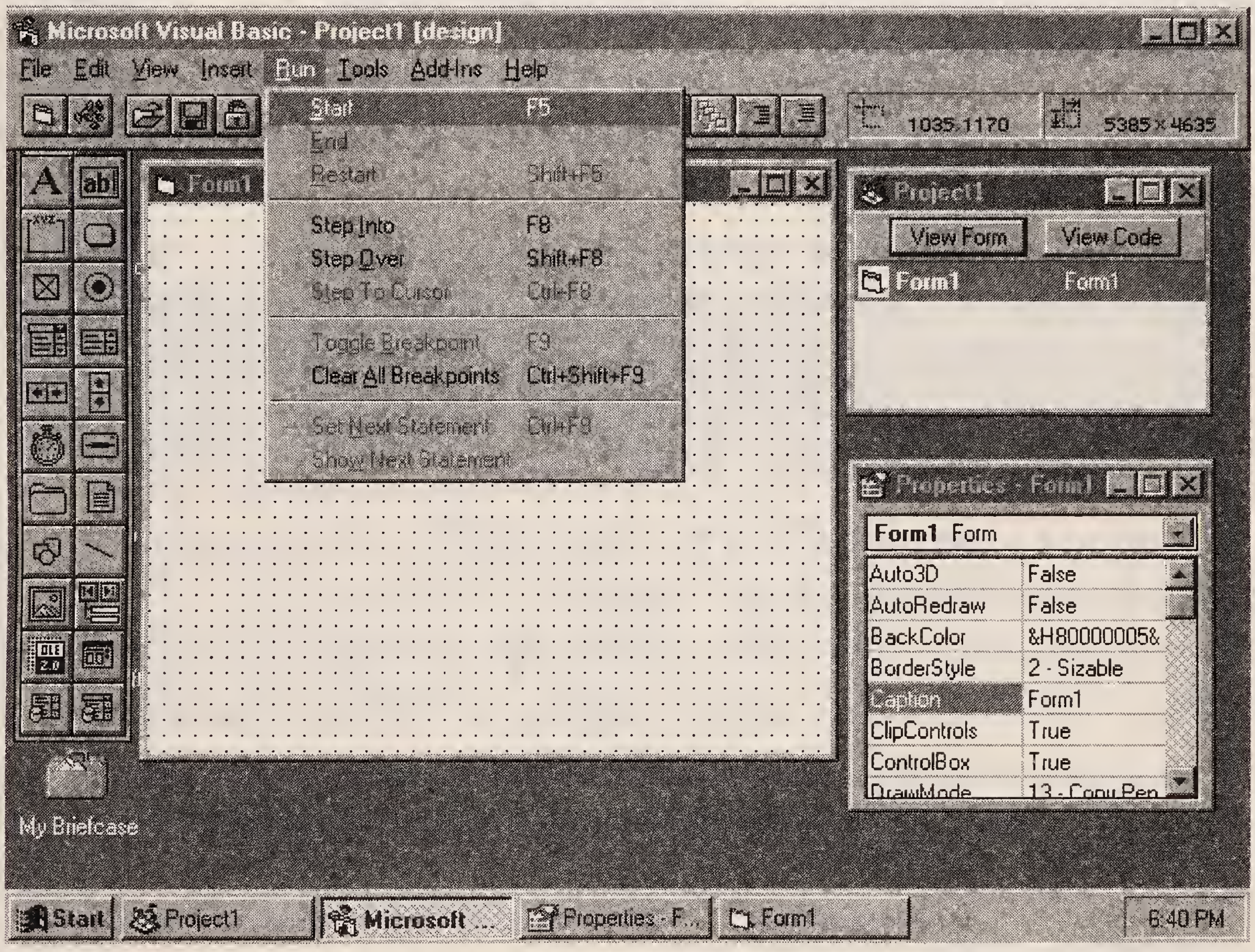
MDI Form 命令用于建立一 MDI 窗体窗口。并把它加到工程文件中。由于一个工程文件只能含有一个 MDI 窗体，所以当工程文件中已含有一 MDI 窗体时，这一命令是无效的。

Module 命令将在某工程文件中插入一新的标准模块。也可以利用工具栏中的 Module（模块）按钮来执行这一命令。

Class Module 命令将在某一工程文件中插入一新的类模块。

（5）Run（运行）菜单

Run 菜单如图 3.19 所示，它由 10 条命令组成，分别是 Start（开始）、End（结束）、Restart（重新启动）、Step Into（步进）、Step Over（单步执行）、Step to Cursor（执行到鼠标）、Toggle Breakpoint（设置断点）、Clear All Breakpoint（清除所有断点）、Set Next Statement（设置下一语句）、Show Next Statement（显示下一语句）。


Microsoft Visual Basic - Project1 [design]
File
Edit
View
Insert
Run
Tools
Add-Ins
Help
Start F5
End
Restart Shift+F5
Step Into F8
Step Over Shift+F8
Step To Cursor Ctrl+F8
Toggle Breakpoint F9
Clear All Breakpoints Ctrl+Shift+F9
Set Next Statement Ctrl+F9
Show Next Statement
1035,1170
5385 x 4635
Form1
Project1
View Form
View Code
Form1 Form1
Properties - Form1
Form1 Form
My Briefcase
Start
Project1
Microsoft ...
Properties - F...
Form1
6:40 PM

Auto3D	False
AutoRedraw	False
BackColor	&H80000005&
BorderStyle	2 - Sizable
Caption	Form1
ClipControls	True
ControlBox	True
DrawMode	13 - Copy Pen

图 3.19　Run（运行）菜单

Start 命令用于启动运行应用程序，同时所有的窗体窗口将被关闭，变量赋初值，从第一个窗体开始运行。当程序在运行过程中，单击 Start 将执行暂停命令，再单击 Start 将继续运行。快捷键是 F5，也可使用工具栏中的 Start 按钮。

Break 命令用于暂停一个正在运行的程序。当执行这一命令时，Visual Basic 将打开调试窗口。被暂停执行的语句将被一由细线组成的框架包围起来，并且被用户在 Environment Options 对话框中选定的一高亮度的颜色覆盖。如果程序处于一个循环状态，正等待一个事件的发生（即没有语句被执行）时，则没有语句被高亮度颜色覆盖。如果在暂停模式下程序被改动了，则需要用户重新启动（Restart）这个程序。快捷键是 F5，Ctrl＋Break 或工具栏中的 Pause 命令。

End 命令用于结束一个程序的运行，也可选择工具栏的 End 按钮。

Restart 命令用于在任何干扰情况下重新运行程序。干扰可能是由运行时的错误引起的、程序中的 Stop 语句引起的、代码中的断点引起的，或者由于选择了 Break（暂停）命令引起的。在这些情况下都可以由这一命令重新运行程序。快捷键是 Shift＋F5，或选择工具栏中

的 Start(开始)按钮。

Step Into 命令用于执行当前的一代码语句。如果这一句的内容是调用一个过程,则下一句将在这一过程中显示被调用的过程中的第一行语句。在设计时,这一命令将在第一段代码被执行以前开始运行并进入暂停模式。程序运行时这一命令是无效的。快捷键是 F8,工具栏中是 Step Into 按钮。

Step Over 命令与 Step Into 相似。只是当被执行语句的内容是调用过程时将有所不同。Step Over 命令按一个单元来执行这一语句,然后进入下一语句。所以,不管当前语句是否调用一过程,下一 语句显示的仍然是当前模块中的下一语句,即被执行的那一语句的下一语句。快捷键是 Shift+F8,工具栏中是 Step Over 按钮。

当应用程序正处于暂停模式时,可以用 Step To Cursor 命令选择一句离当前光标较远的语句,作为程序运行的终点(Stop)。用户可以运行这一命令来避免一个很大的循环。快捷键是 Ctrl+F8。

Toggle Breakpoint 命令将设置或删除当前行中的断点(breakpoint)。但用户不可以在一些不是执行语句的代码行,如注释、说明语句或空行上设置断点。被设置断点的语句将被用户在 Format Options 对话框里选择的高亮颜色所覆盖。快捷键是 F9,工具栏中是 Toggle Breakpoint 按钮。

Clear All Breakpoint 命令用于将工程文件中的所有断点全部消除。这样,就可以运行一个完整的应用程序了。当然,如果用户设置了表达式观察(watch expression)或发生错误时暂停项,则程序运行时仍会发生断点现象。快捷键是 Ctrl+Shift+F9。

Set Next Statement 命令用于在当前语句后设置一行不同的代码来运行。效果与 Step Into 命令相似。只是 Step Into 命令只执行了过程中的下一行代码。运用 Step Next Statement(进入下一语句)命令,用户可以选择当前语句的前面或后面的一行代码。任何插入代码都不会被执行。运用这一命令,可以在当前过程重新运行一段代码或者浏览一下用户想要执行的语句。但不能用 Step Next Statement 命令时进入不同过程中的语句。这一命令只有在暂停模式下才是有效的。快捷键是 Ctrl+F9。

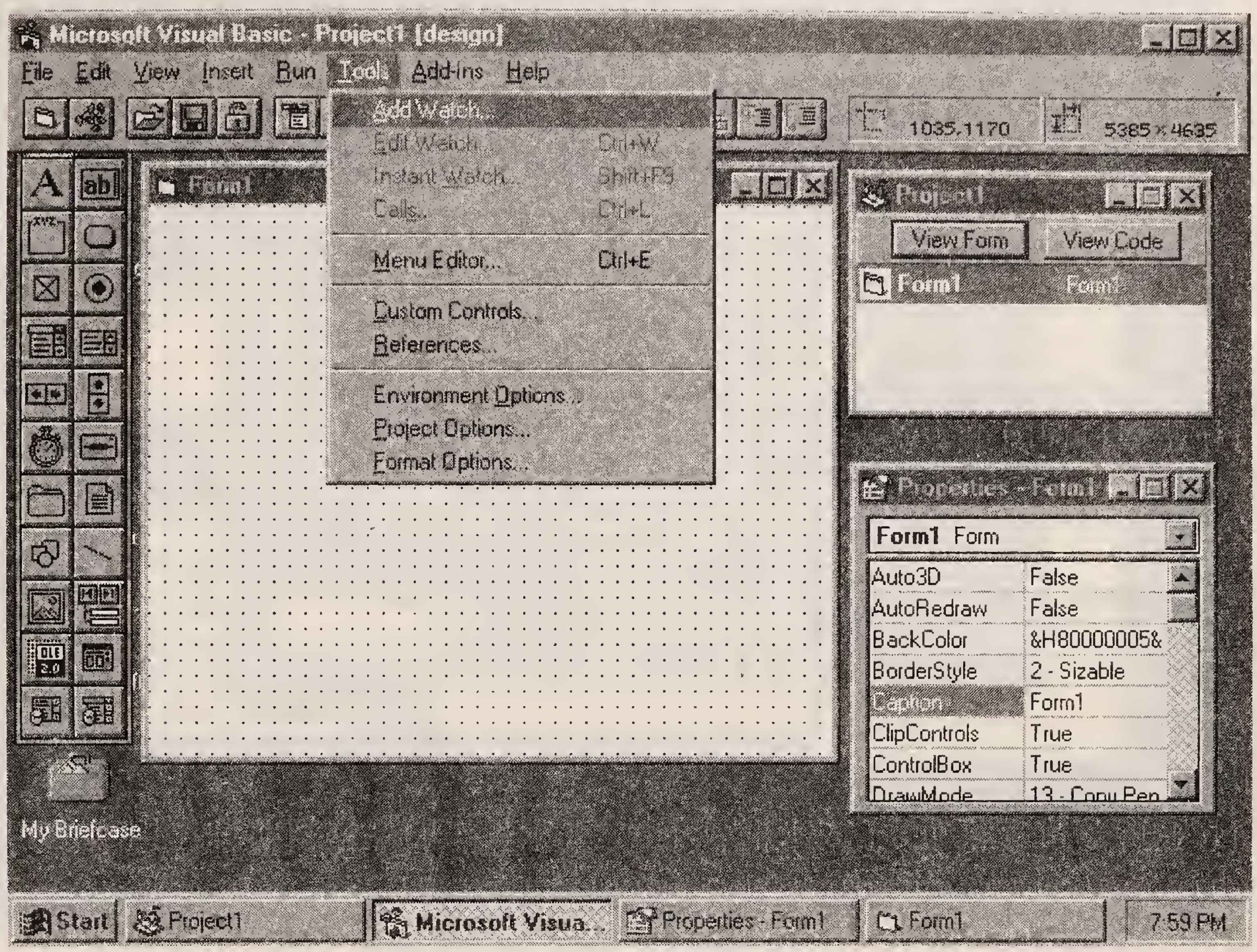
Show Next Statement 命令用于将光标放在将要执行的语句前面。

(6) Tools(工具)菜单

Tools(工具)菜单,如图 3.20 所示。共有 10 条命令,分别是 Add Watch(加入观察变量)、Edit Watch(对观察变量进行编辑)、Instant Watch(即时观察)、Calls(调用)、Menu Editor(菜单编辑器)、Custom Control(定制控件)、References(引用)、Environment Options(环境选项)、Project Options(工程文件选项)、Format Options(格式选项)。

Add Watch 命令,在设计或暂停模式下,这一命令将显示 Add Watch(加入观察)对话框,这样就可以观察变量的当前值(watch expression)。变量可以是任何合法的 Basic 语言表达式。如果调试窗口是可视的,则每当暂停时,就可以在调试窗口的 Watch 观察板上观察变量的值了。

Edit Watch 命令,显示 Watch Edit 对话框,用户可以修改删除一条观察变量。快捷键是 Ctrl+W。


Microsoft Visual Basic - Project1 [design]
File
Edit
View
Insert
Run
Tools
Add-Ins
Help
1035,1170
5385 x 4635
Add Watch...
Edit Watch...　Ctrl+W
Instant Watch...　Shift+F9
Calls...　Ctrl+L
Menu Editor...　Ctrl+E
Custom Controls...
References...
Environment Options...
Project Options...
Format Options...
Form1
Project1
View Form
View Code
Form1　Form1
Properties - Form1
Form1 Form
My Briefcase
Start
Project1
Microsoft Visua...
Properties - Form1
Form1
7:59 PM

Auto3D　False
AutoRedraw　False
BackColor　&H80000005&
BorderStyle　2 - Sizable
Caption　Form1
ClipControls　True
ControlBox　True
DrawMode　13 - Copy Pen

图 3.20　Tools(工具)菜单

Instant Watch，显示 Instant Watch 对话框。当对程序进行调试时，用一命令可以观察变量、属性的当前值或没有被定义的变量的当前值。选择代码窗口或调试窗口代码板中的变量，然后选择 Instant Watch 命令即可。快捷键是 Shift＋F9 或工具栏中的 Instant Wath 命令。

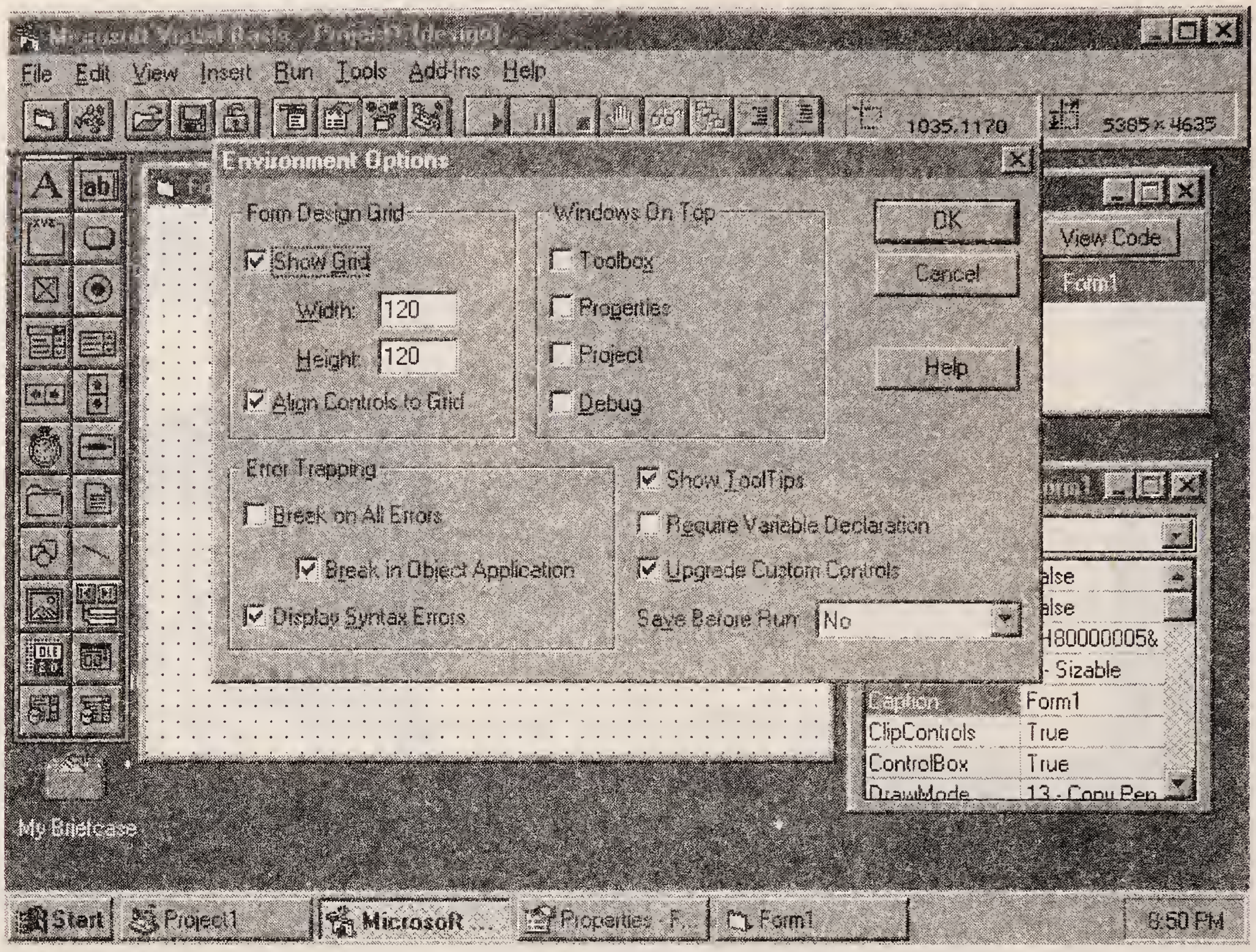
Calls 命令，显示调用对话框，显示当前活动的调用过程，即在应用程序还没有运行完毕的调用这程。这一命令只在暂停模式下有效。如果 Visual Basic 正在执行一个过程的代码，那么这一过程将被列到当前活动过程目录里。如果这一过程正在调用另一过程，则这两个过程同时被列在目录里。从目录里删除的正在运行的过程将回到调用程序那里去。从调试窗口里调用的程序也将列在目录里。快捷键是 Ctrl＋L，工具栏里是 Calls 命令。

Menu Editor 命令，打开菜单编辑窗口，快捷键是 Ctrl＋E 或工具栏里的 Menu Editor 按钮。

Custom Controls 命令用于显示定制控件对话框，允许用户把一个 VBX 定制控件，一个 OLE(对象链接与嵌入)定制控件或一个可插入对象(insertable object)添加到工具栏中。

References，显示一个 References(引用)对话框，允许用户把一个对象库或一个类型库(type library)添加到工程文件中，使得用户可以使用其他应用程序中的对象。当 References 被设置以后，应用程序的对象将被显示在对象浏览器中。

Environment Options 显示一个 Environment Options 对话框，允许用户对 Visual Basic 程序设计环境进行设置。如图 3.21 所示。


Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
1035,1170
5385 x 4635
Environment Options
OK
Cancel
Help
View Code
Form1
My Briefcase
Start
Project1
Microsoft ...
Properties - F...
Form1
6:50 PM

Form Design Grid

☑ Show Grid

Width: 120

Height: 120

☑ Align Controls to Grid

Windows On Top

☐ Toolbox

☐ Properties

☐ Project

☐ Debug

Error Trapping

☐ Break on All Errors

☑ Break in Object Application

☑ Display Syntax Errors

☑ Show ToolTips

☐ Require Variable Declaration

☑ Upgrade Custom Controls

Save Before Run: No

Caption	Form1
ClipControls	True
ControlBox	True

图 3.21 Environment Options(环境选项)对话框

Project，显示工程文件窗口，快捷键是 Ctrl+R，或工具栏中的 Project 命令。

Format Options，显示一 Format Options 对话框，可以对字体、字体尺寸以及代码字体颜色等进行设置。

(7) Add-Ins(附件)菜单

Add-Ins 菜单如图 3.22 所示，它共有 3 条命令，分别是 Data Manager(数据管理器)，Report Designer(报表设计器)，Add-In Manager(附件管理器)。

Data Manager 打开或启动一数据管理器。

Report Designer 打开或激活 Visual Basic 中的 Crystal Reports 工具，这个命令存在于 Visual Basic 专业版上，并且只有在安装时装上 Crystal Reports 才有效。

Add-In Manage，打开用于扩展 Visual Basic 开发环境的附件管理器。

(8) Help(帮助)菜单

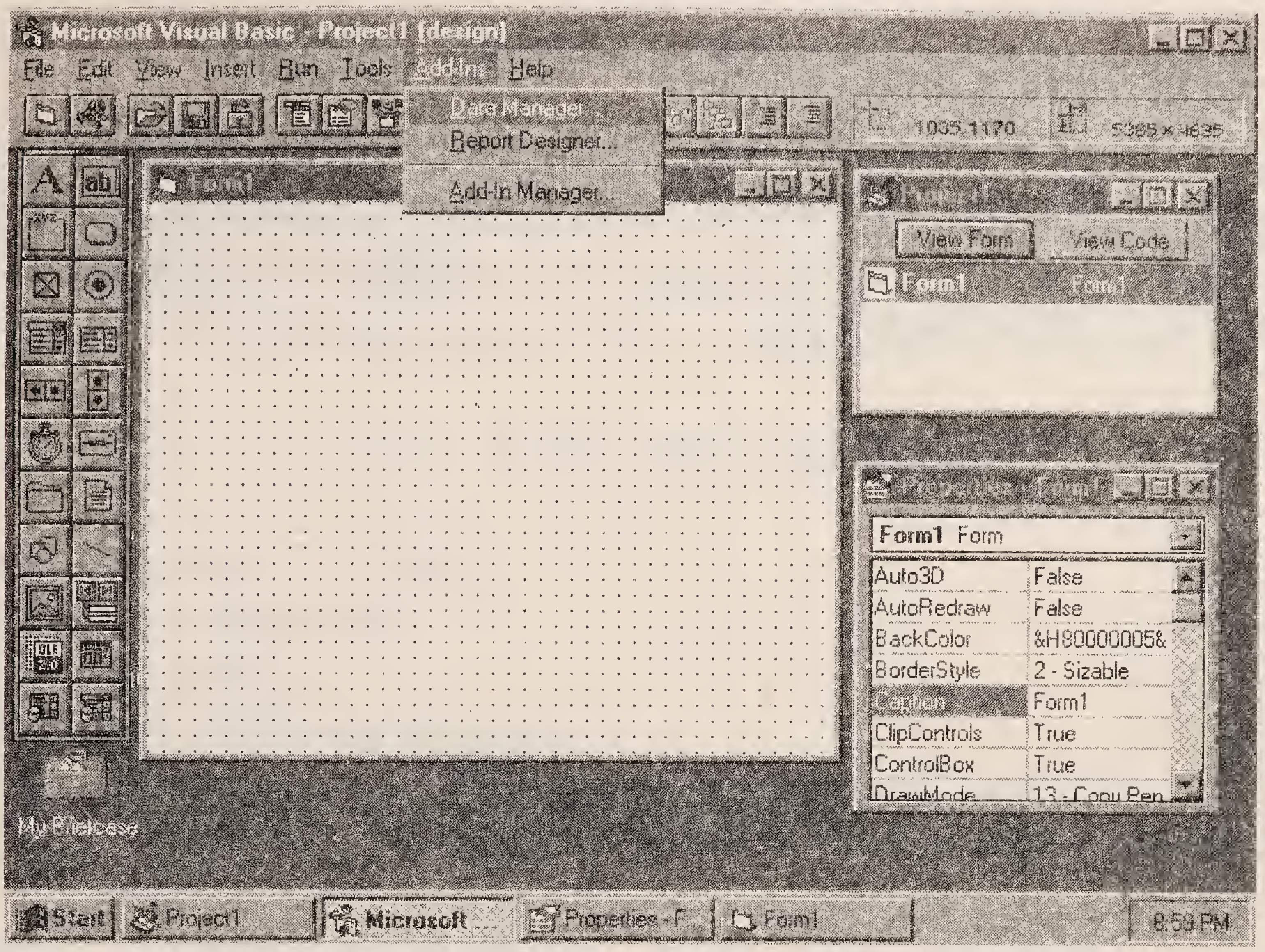
Help(帮助)菜单如图 3.23 所示，共由 5 条命令组成，分别是 Contents(目录)、Search Help On(寻找帮助)、Obtaining Technical Support(获得技术帮助)、Learning Microsoft Visual Basic(学习 Microsoft Visual Basic)、About Microsoft Visual Basic(关于 Microsoft Visual Basic 的一些信息)。

Contents 命令运行 Help 并且显示有关 Visual Basic 4.0 内容的目录。

Search For Help On 命令显示寻找帮助内容的对话框，使用户可以快速找到需要的帮助信息。

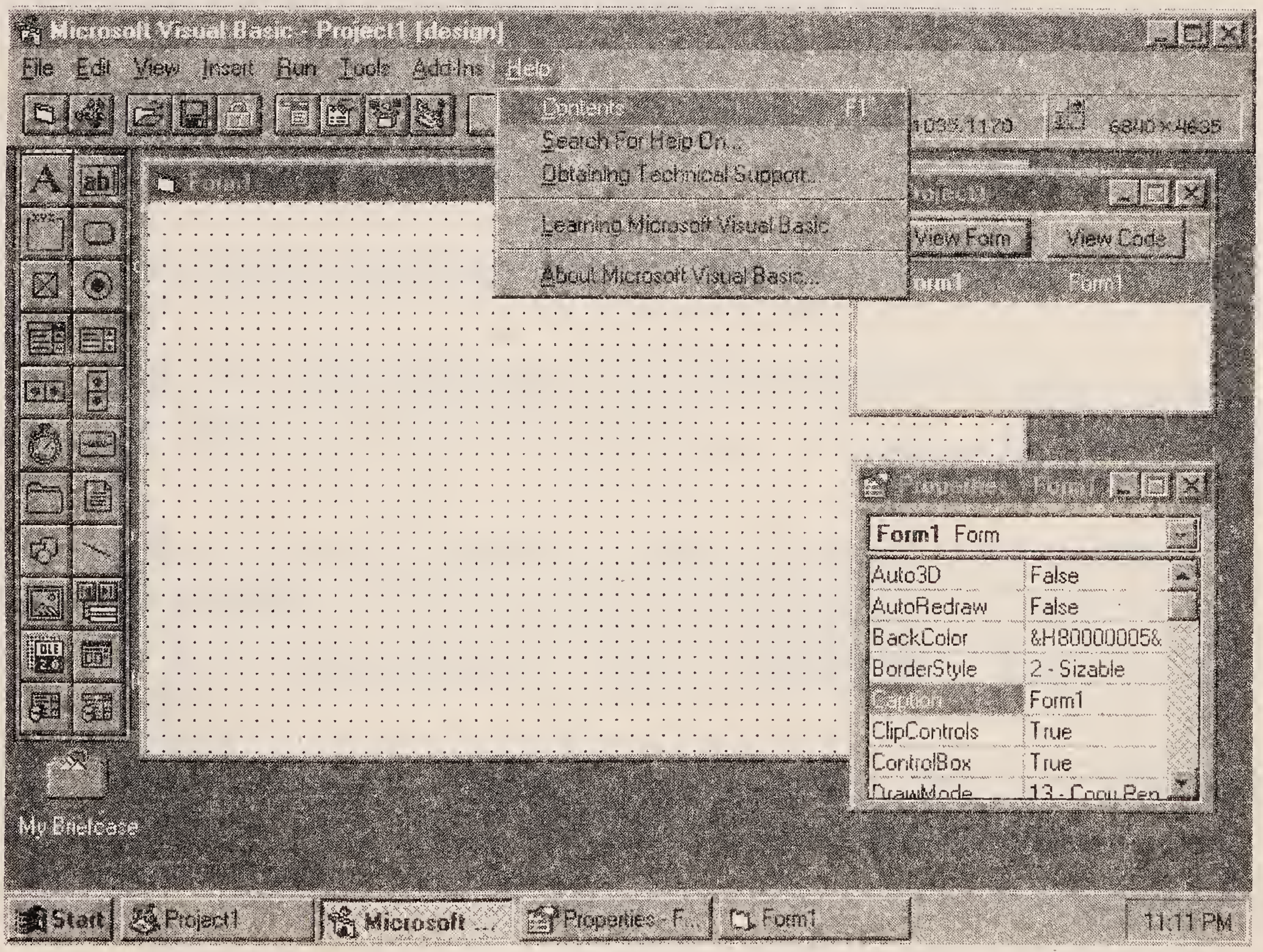
Obtaining Technical Support 命令，运行 Help 并且显示有关 Microsoft 产品支持设备的内容。

Learning Microsoft Visual Basic 命令让用户学习怎样使用 Visual Basic。


Microsoft Visual Basic - Project1 [design]
File
Edit
View
Insert
Run
Tools
Add-Ins
Help
Data Manager...
Report Designer...
Add-In Manager...
1035,1170
5385 x 4635
Form1
Project1
View Form
View Code
Form1
Form1
Properties - Form1
My Briefcase
Start
Project1
Microsoft ...
Properties - F...
Form1
8:59 PM

Form1 Form
Auto3D False
AutoRedraw False
BackColor &H80000005&
BorderStyle 2 - Sizable
Caption Form1
ClipControls True
ControlBox True
DrawMode 13 - Copy Pen

图 3.22 And-Ins(附件)菜单


Microsoft Visual Basic - Project1 [design]
File
Edit
View
Insert
Run
Tools
Add-Ins
Help
Contents F1
Search For Help On...
Obtaining Technical Support...
Learning Microsoft Visual Basic
About Microsoft Visual Basic...
1035,1170
6840 x 4635
Form1
Project1
View Form
View Code
Form1
Form1
Properties - Form1
My Briefcase
Start
Project1
Microsoft ...
Properties - F...
Form1
11:11 PM

Form1 Form
Auto3D False
AutoRedraw False
BackColor &H80000005&
BorderStyle 2 - Sizable
Caption Form1
ClipControls True
ControlBox True
DrawMode 13 - Copy Pen

图 3.23 Help(帮助)菜单

About Microsoft Visual Basic 显示一对话框，介绍有关版本、拷贝注意事项以及与之相关的内容。

3.2.4 标题条(Title Bar)

最后我们再来介绍一下在 Visual Basic 启动后工作界面的标题条。标题条的最左边是一图标,用鼠标单击时会弹出一控件菜单,用于移动、改变界面大小、缩为图标、关闭窗口等。图标右边是标题 Microsoft Visual Basic-Project1[design],表示现在正在对工程文件进行设计。当运行程序时,[]内的内容将变为 run。

第四章　建立一个简单的应用程序

前面几章，我们简单地介绍了 Windows 95，Visual Basic 4.0 的一些知识，也介绍了 Visual Basic 的编程环境。下面我们就介绍一个简单的程序，这一程序虽没有多大意义，但可以使读者对建立一个应用程序的步骤和方法，有一个大概的了解。

4.1　简单应用程序的设计

建立一个应用程序分三步进行：

(1) 建立用户界面的对象。

(2) 为对象设置属性。

(3) 把程序代码加到对象的控制中去。

下面我们就按照这三步建立一个简单的能进行加法运算的应用程序。

4.1.1　建立用户界面的对象

要建立一个应用程序，首先我们要建立一个新的工程文件。可以通过 File（文件）菜单中的 New Project（新工程文件）命令来建立一个新 Project 和 Form 窗口。

一个工程文件包含三部分的内容：Form（窗体）窗口、Controls（控件）和 Code（代码）窗口。Form 是程序运行时的背景窗口和对话框，控件是放置在窗体中的具体的对象，而代码则是具体的过程与说明。每一个工程文件至少包含一个 Form 窗口，在我们这个程序里，只要一个就可以了。

Form 窗口是应用程序的运行背景，用户可以通过改变它的位置或大小来满足应用程序的设计要求。

下一步，就可以从工具框里选择控件了。在这个程序里，我们一共需要七个控件，分别是三个 Label（标签）控件，三个 Text（文本框）控件以及一个 Command Button（命令按钮）控件。

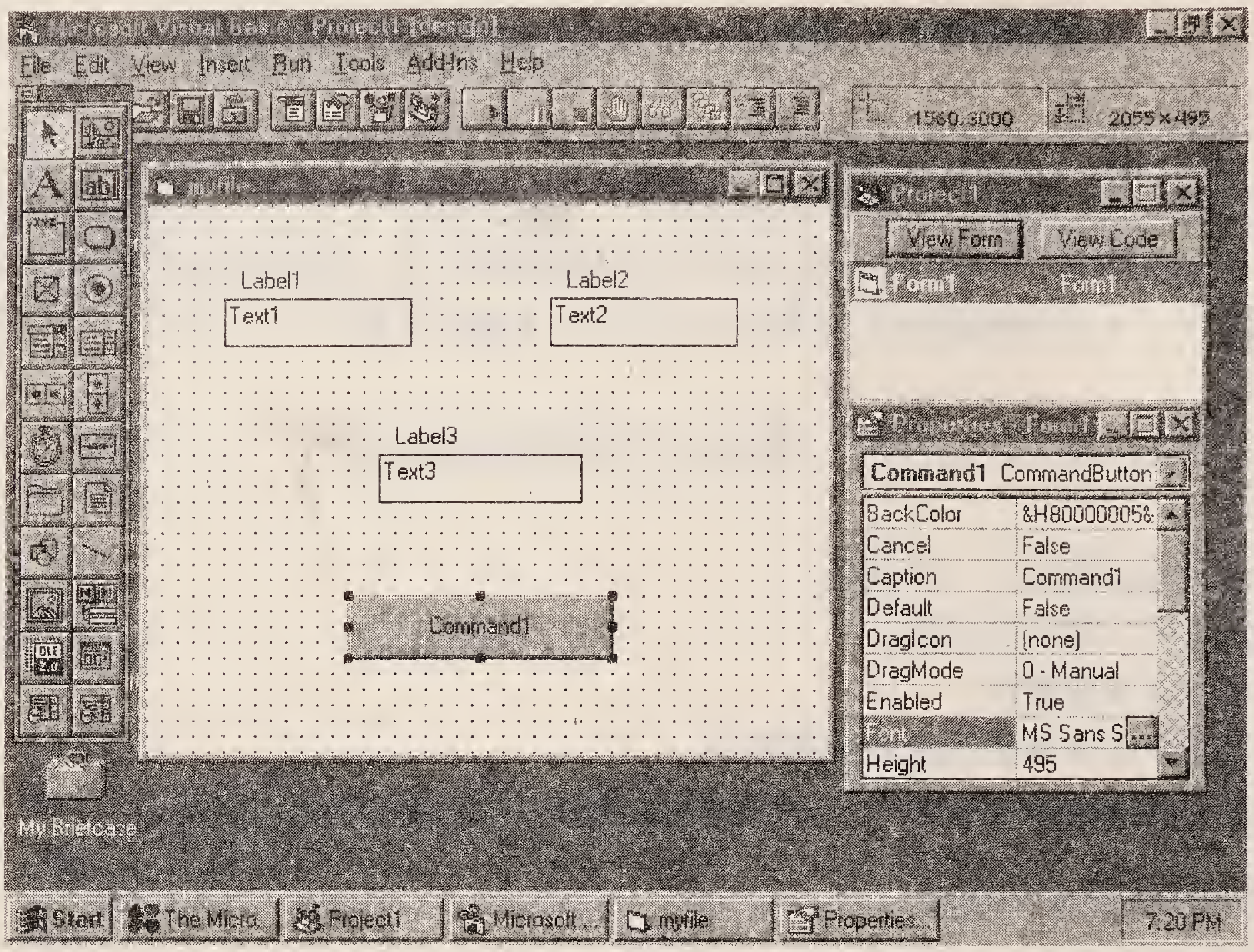
首先，从工具框中选好控件，然后在 Form 窗口上按动鼠标左钮并拖动鼠标，画出一个框体，放开鼠标按钮即绘出一个控件。也可以在系统设置情况下，在工具框中的控件上双击鼠标，即在 Form 窗口上的缺省位置形成一个框体。框体周围的八个黑色小方框是为了改变框体的形状而设置的。

在这个程序里需要设置三个文本框（Text），两个用来输入数据，另一个用来输出数据。当在上面两个框体中输入数据以后，第三个框体将输出两数据相加后的值。

为了表示每个文本框所表达的内容，还需要加入标签（Label）。标签的内容是不能改变的。

最后再加入一个命令按钮（Command Button），当输入两个数据以后，单击命令按钮将输出两个数据之和。

绘制好控件的 Form 窗体如图 4.1 所示。


Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
1560, 3000
2055 x 495
myfile
Label1
Text1
Label2
Text2
Label3
Text3
Command1
Project1
View Form
View Code
Form1
Form1
Properties - Form1
My Briefcase
Start
The Micro...
Project1
Microsoft...
myfile
Properties...
7:20 PM

Command1 CommandButton

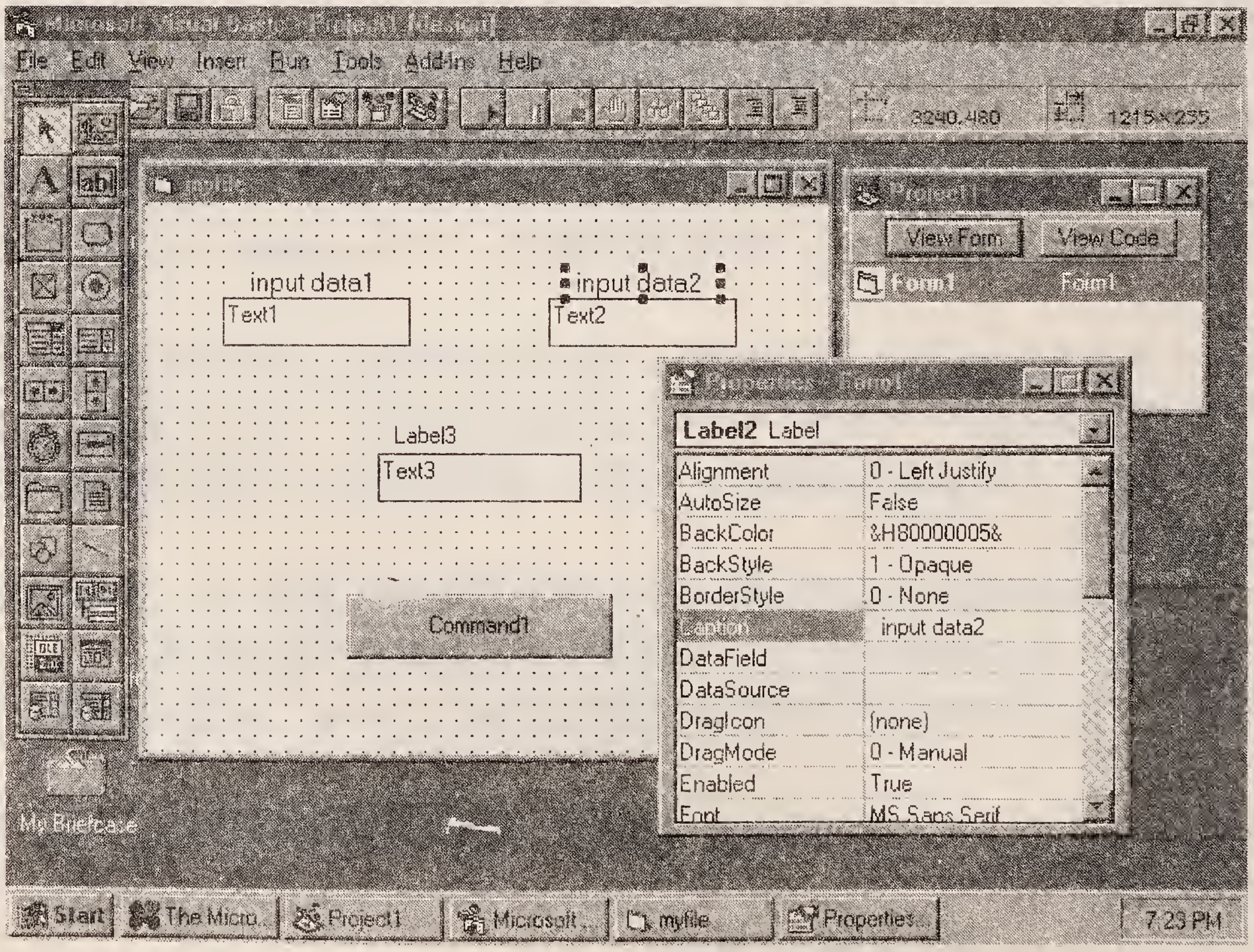
BackColor	&H80000005&
Cancel	False
Caption	Command1
Default	False
DragIcon	(none)
DragMode	0 - Manual
Enabled	True
Font	MS Sans S
Height	495

图 4.1　建立用户界面的对象

4.1.2　为对象设置属性值

对象选择好以后，就要为其设置属性值。

首先，选择一个窗体或控件，则它的属性和设置将显示在属性窗口里，如图 4.2 所示。


Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
3240, 480
1215 x 255
myfile
input data1
Text1
input data2
Text2
Label3
Text3
Command1
Project1
View Form
View Code
Form1
Form1
Properties - Form1
My Briefcase
Start
The Micro...
Project1
Microsoft...
myfile
Properties...
7:23 PM

Label2 Label

Alignment	0 - Left Justify
AutoSize	False
BackColor	&H80000005&
BackStyle	1 - Opaque
BorderStyle	0 - None
Caption	input data2
DataField	
DataSource	
DragIcon	(none)
DragMode	0 - Manual
Enabled	True
Font	MS Sans Serif

图 4.2　选择控件并为其设置属性

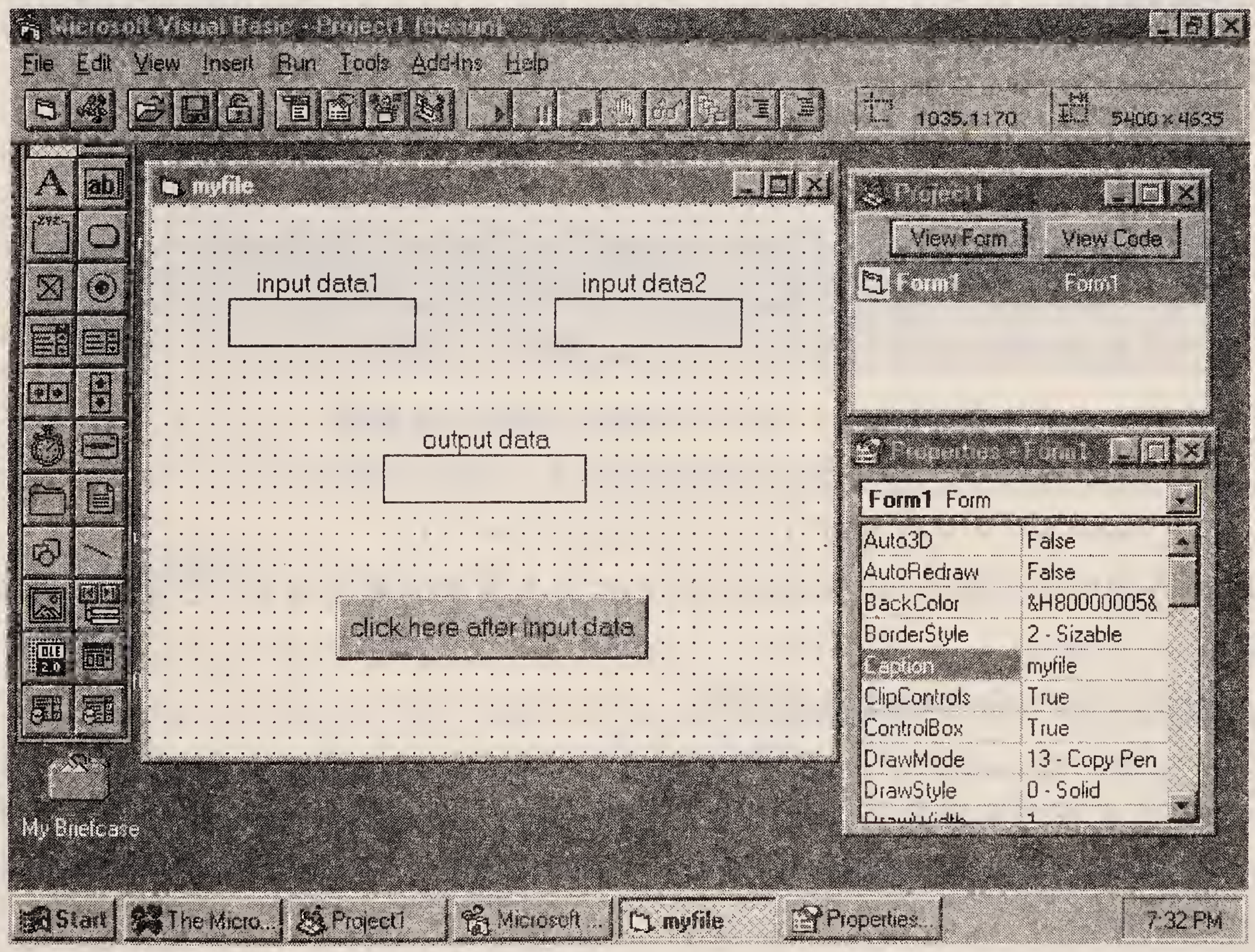
对于 Form1（窗体）窗口，我们选择它的 Caption（标题）属性为 my file（我的文件）。对于文本框 1（Text1），我们选择它的控件名（Name）属性，这一属性在代码中将代表文本框 1，将其设置为 data1（数据 1）。对于文本框 2（Text2），我们将其 Name 属性设置为 data2（数据 2）。对于文本框 3（Text3），将其 Name 属性设置为 data3（数据 3）。

对于标签 1（Label1），我们选择它的标题（Caption）属性。这一属性将确定在这一控件中选定的文本内容。由于 data1 和 data2 文本框的内容为输入的数据，而 data3 的内容为输出的数据，所以我们给 Label1 设置的 Caption 属性为 input data1（输入数据 1），Label2 的 Caption 属性设置为 input data2（输入数据 2），Label3 的 Coption 属性设置为 output data（输出数据）。

对于命令按钮（Command Button），我们选择它的标题（Caption）属性。由于它执行的命令是单击后即输出数据 1 与数据 2 的和，所以我们取它的 Capiton 属性为 click here after input data（输入数据后单击此处）。再选取（Name）属性，将其设置为 add。

对于文本框 1、文本框 2、文本框 3，我们还要选择它们的文本（Text）属性，将它们设置为空，即清除文本框中的内容，以便于数据的输入与输出。

属性设置好后的 Form 窗口如图 4.3 所示。


Microsoft Visual Basic - Project1 [design]
File
Edit
View
Insert
Run
Tools
Add-Ins
Help
1035,1170
5400 x 4635
myfile
input data1
input data2
output data
click here after input data
Project1
View Form
View Code
Form1
Form1
Properties - Form1
My Briefcase
Start
The Micro...
Project1
Microsoft ...
myfile
Properties...
7:32 PM

Form1 Form

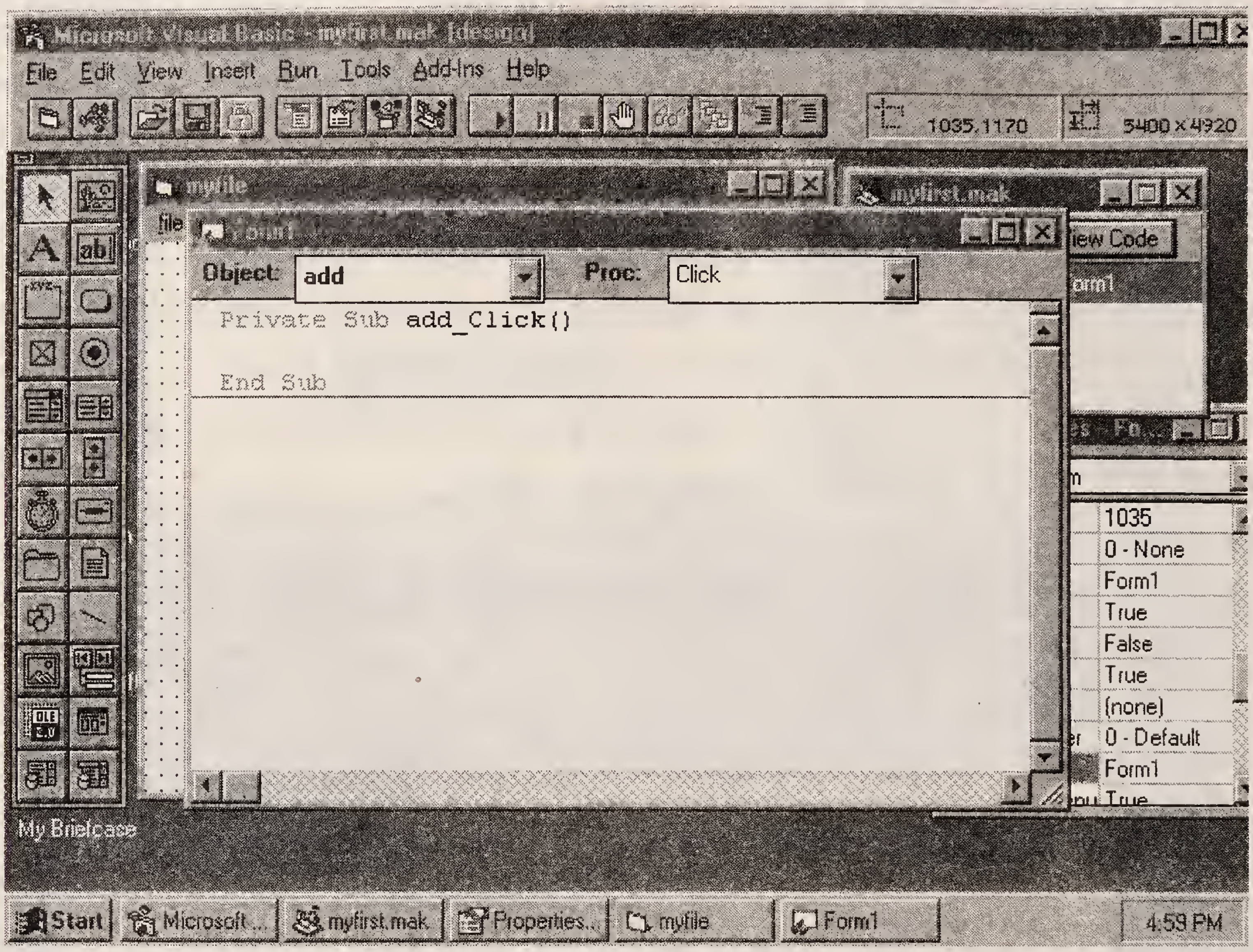
Auto3D	False
AutoRedraw	False
BackColor	&H80000005&
BorderStyle	2 - Sizable
Caption	myfile
ClipControls	True
ControlBox	True
DrawMode	13 - Copy Pen
DrawStyle	0 - Solid
DrawWidth	1

图 4.3　属性设置好后的 Form（窗体）窗口

4.1.3　把程序代码加到对象的控件中去

选定对象并对其属性设置以后的工作就是对其进行编程。编程总是在 Code 窗口中进行。在 Form 窗口中双击鼠标，或单击 Project 窗口中的 View Code 按钮，即可进入 Code 窗口。

用鼠标单击 Object（对象）列表框右边的箭头按钮，即拉出一列表框，列出了这个应用程序所包含的所有对象。选择 add。然后用鼠标单击 Procedure（过程）列表框右边的按钮，拉出一列表框，列出与 add 对象相关的所有事件，从中选择 Click。此时，Code 窗口中就显示出 add _ Click 事件代码的过程头与过程尾如图 4.4 所示。


Microsoft Visual Basic - myfirst.mak [design]
File Edit View Insert Run Tools Add-Ins Help
1035,1170
5400 x 4920
myfile
myfirst.mak
View Code
Form1
My Briefcase
Start
Microsoft ...
myfirst.mak
Properties...
myfile
Form1
4:59 PM

Object: add　Proc: Click

```
Private Sub add_Click()

End Sub
```

图 4.4　add _ Click 事件过程头与过程尾

在 add _ Click 事件的过程头与过程尾之间输入下面的代码：

```
data3.Text=Str $(Val(data1.Text)+Val(data2.Text1))
```

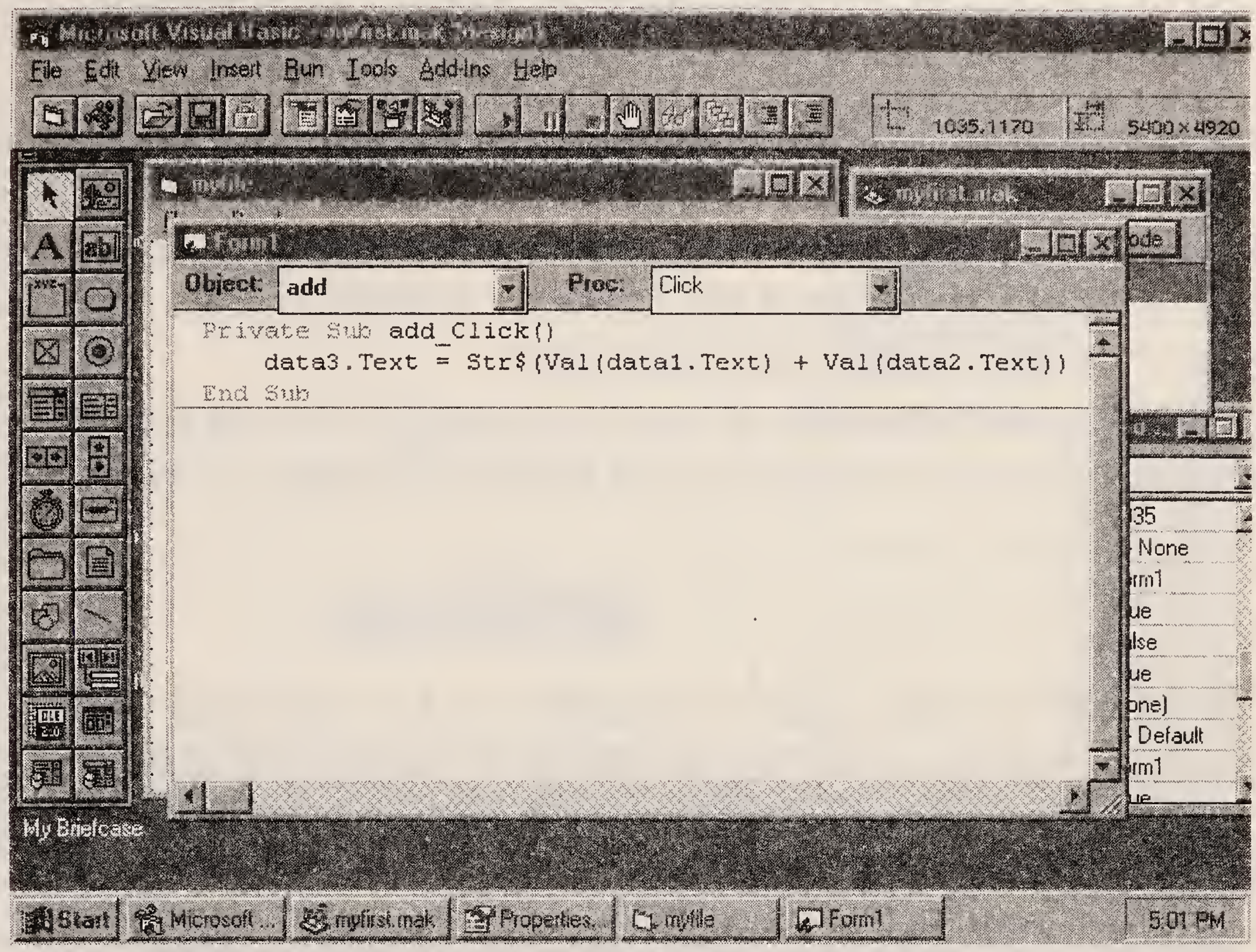
这段代码的意思是将 data1 文本框与 data2 文本框中的数据相加，把结果在 data3 文本框中输出。

在这里我们看到，程序中没有直接写：

```
data3.Text=data1.Text+data2.Text
```

这是因为，文本框中的数字为数字型字符串，需由 Val(string $) 函数将其转换为数字，才能进行运算，运算后的结果仍为数字，在文本框中输出时，还需将数字由 Str $(number) 函数转换为字符串。

此时代码窗口如图 4.5 所示。由一个事件过程组成，就是单击命令按钮事件。这样，一个完整的加法应用程序就设计完成了。


Microsoft Visual Basic - myfirst.mak [design]
File Edit View Insert Run Tools Add-Ins Help
1035,1170
5400×4920
myfile
myfirst.mak
Form1
My Briefcase
Start
Microsoft ...
myfirst.mak
Properties...
myfile
Form1
5:01 PM

Object: add　Proc: Click

```
Private Sub add_Click()
    data3.Text = Str$(Val(data1.Text) + Val(data2.Text))
End Sub
```

图 4.5　包含一个事件过程的代码窗口

4.2　掌握运行模式的概念

在程序设计完成以后，就要运行程序。在运行以前，大家有必要掌握一下运行模式的概念。通常，程序可以在两种模式下运行，分别是解释运行模式和编译运行模式。

4.2.1　解释运行模式

当用户选择了 Run 菜单中的 Start 命令或单击工具栏中的 Start 按钮时，Visual Basic 解释程序就投入运行，并开始运行用户程序。首先，用于生成应用程序的 Form 和 Project 窗口将被关闭；其次，解释程序显示用户程序中定义的第一个 Form（窗体），然后解释程序逐行开始运行用户程序，解释器同时将 Debug 窗口打开。

解释器每读完一行代码，就将其转换为机器代码，然后再执行这些命令。机器代码是微处理器能理解的数字序列。因为微处理器指令并未保存，一行代码执行以后，如需再次运行，就必须再解释一次。这种运行方式的好处是当对一个程序作了修改以后，不必编译就可以立即执行。但由于每条语句在执行前必须被解释一次，这种运行方式比编译程序慢得多。

在程序运行过程中，用户不能修改程序代码，也不能打开或保存工程文件。如果执行 Break 命令，则 Visual Basic 会暂停程序执行，并弹出 Debug 窗口。在此状态中，允许用户修改代码，并从停止处重新运行程序。如果程序执行完毕，用户选择 Run 菜单中的 End 命令或单击工具栏中的 End 按钮，则 Visual Basic 又回到 Design 模式中。如果用户是在 Form 窗口启动运行程序的，则结束后重又出现 Form 窗口。如是在 Code 窗口启动运行程序的，则结束后重新回到 Code 窗口。

4.2.2　编译运行模式

编译运行模式也是应用程序的一种运行模式，但它实际上并不是 Visual Basic 真正的模式。在编译一个程序时，Visual Basic 读取程序中的语句，解释这些语句，并把它们转换成微处理器命令，然后将这些微处理器命令保存于 EXE（可执行文件）当中。因为程序中所有的语句都已经被解释并转换为微处理器命令，这样在执行程序时就不必解释一句执行一句了，这样就节省了一个步骤。因而运行速度也相应提高了。

在执行 EXE 文件时，微处理器不需 Visual Basic 即可读取并执行微处理器命令，只是程序必须能够存取 Visual Basic 的实时运行库，这样就可以独立地执行一个程序了。

4.3　运行程序

下面就从解释方式下运行上面设计的加法程序。从 Run 菜单中选择 Start 命令或单击工具栏中的 Start 按钮时，Code 窗口或 Form 窗口以及 Properties 窗口都关闭了。并出现如图 4.6 所示的一个 Form 运行窗口，以及 Debug 窗口。一般，Debug 窗口处于 Form 运行窗口下面，要想进行调试，只要在 Debug 窗口上单击一下即可。

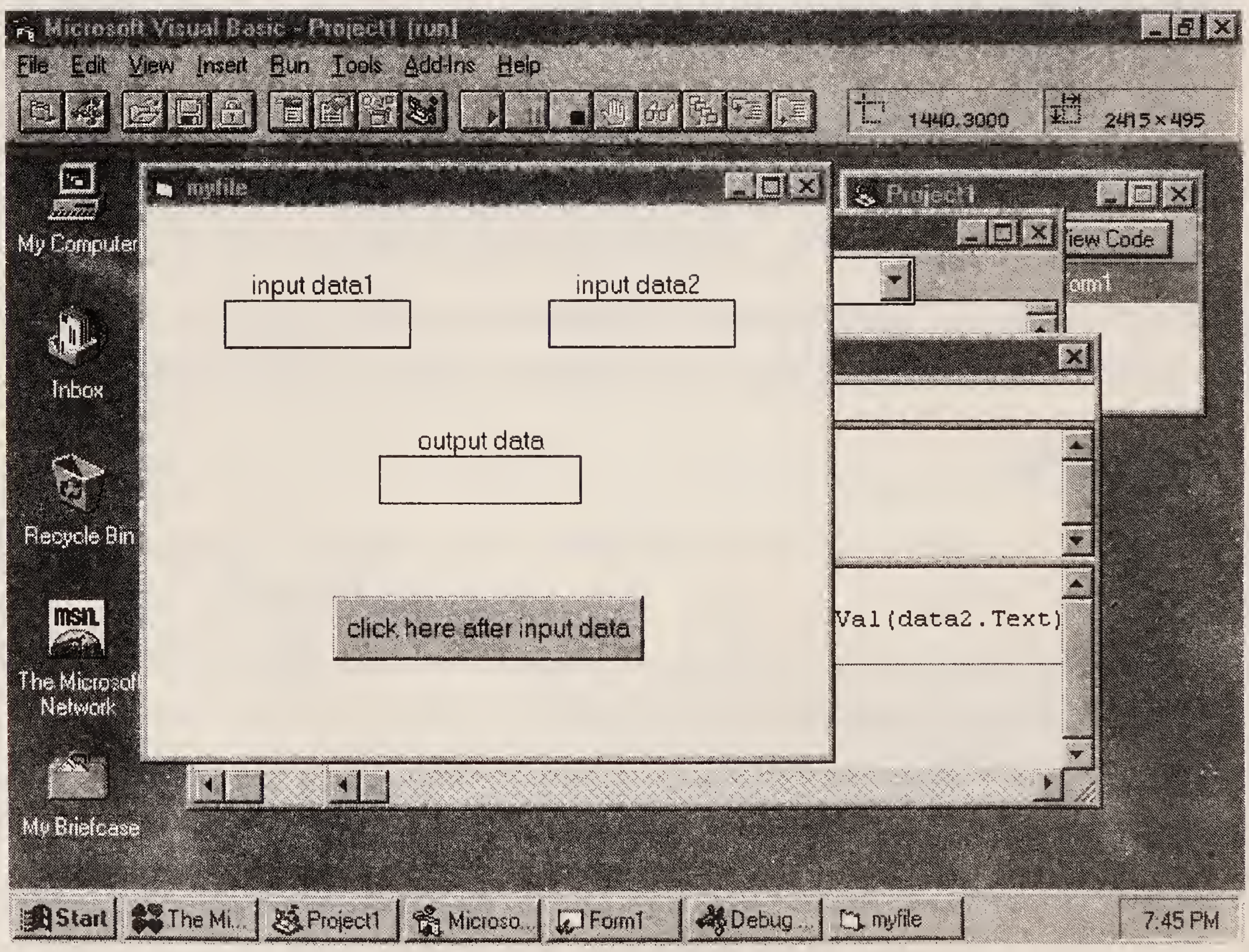
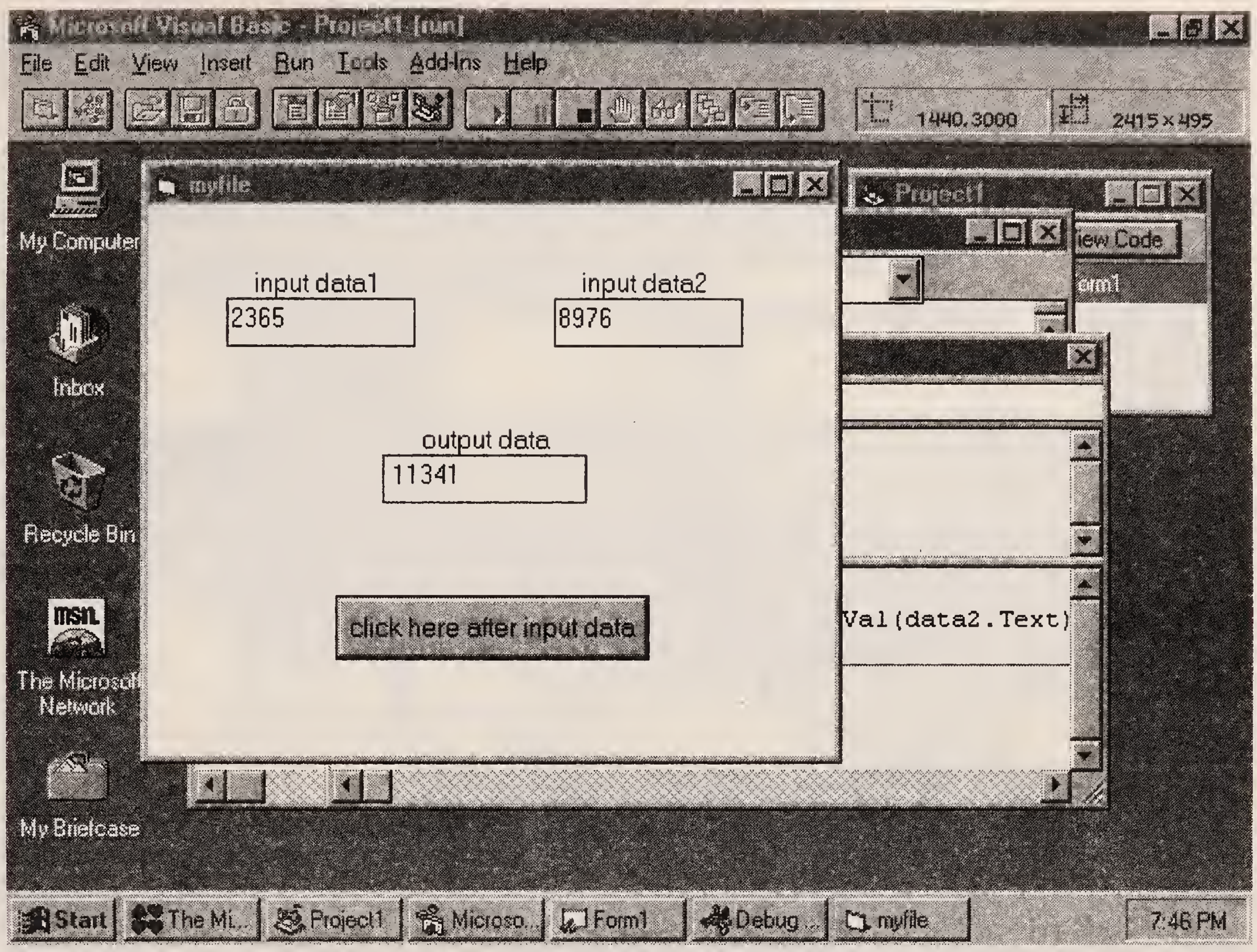

Microsoft Visual Basic - Project1 [run]
File Edit View Insert Run Tools Add-Ins Help
1440,3000
2415×495
myfile
input data1
input data2
output data
click here after input data
Project1
View Code
orm1
Val(data2.Text)
My Computer
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Start
The Mi...
Project1
Microso...
Form1
Debug...
myfile
7:45 PM

图 4.6　启动程序后屏幕

在 input data1 标签下的文本框中会看到一活动光标，表示可由这里输入数据。键入一整数后，单击 Tab 键，则活动光标移到 input data2 标签下的文本框中，在这一文本框中输入另一数值。然后单击命令按钮，则在 output data 标签下的文本框中输出前两个输入数据的和。如图 4.7 所示。


Microsoft Visual Basic - Project1 [run]
File Edit View Insert Run Tools Add-Ins Help
1440,3000
2415×495
myfile
Project1
View Code
Form1
My Computer
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
click here after input data
Start
The Mi...
Project1
Microso...
Form1
Debug...
myfile
7:46 PM

input data1: 2365

input data2: 8976

output data: 11341

```
Val(data2.Text)
```

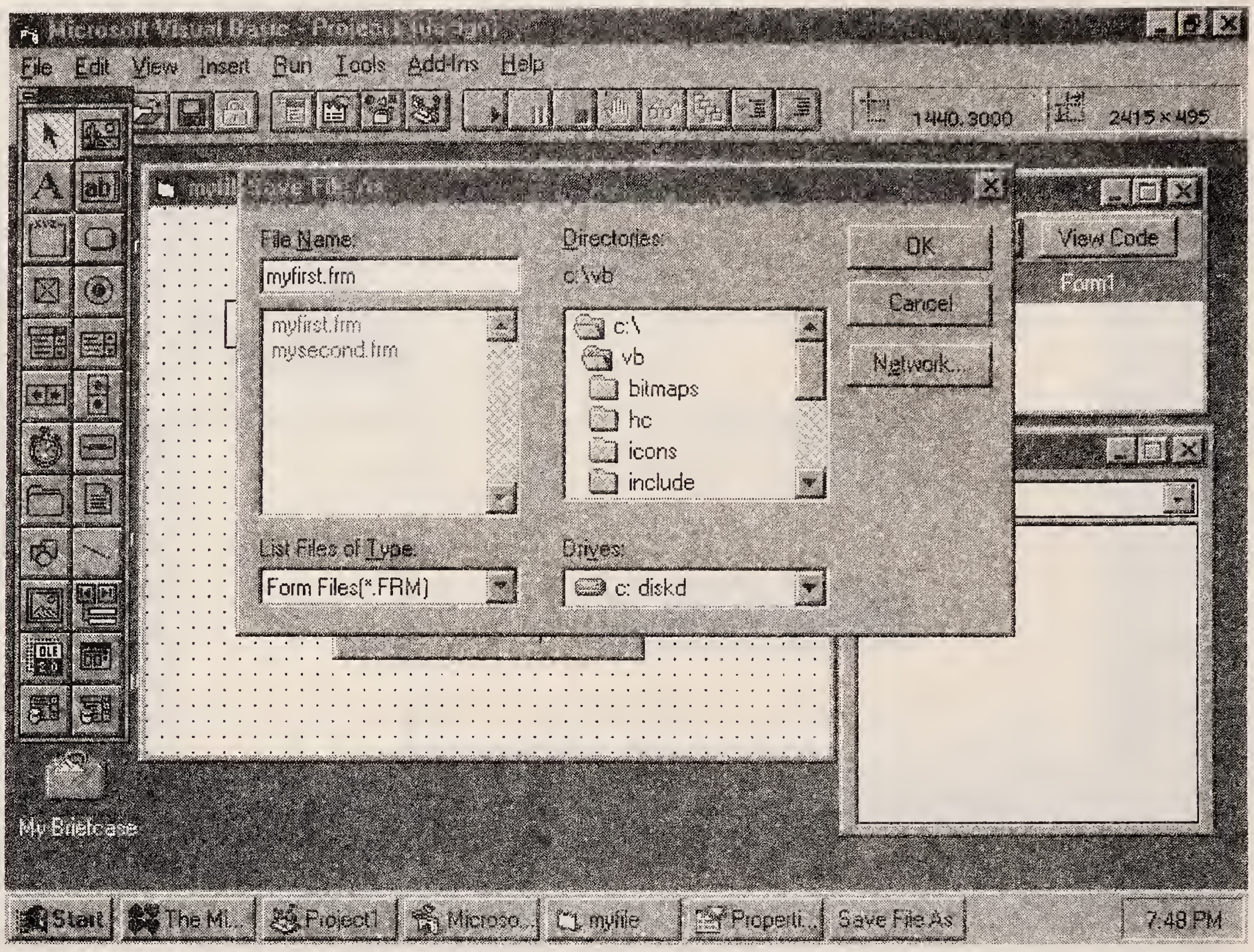
图 4.7　程序运行结果

如果程序能够运行，但发现结果不对，可以执行 Run 菜单中的 End 命令或单击工具栏中的 End 按钮，回到 Code 窗口，对程序及属性的设置进行检查，直到发现错误为止。

4.4　保存文件

程序运行结束以后，确保运行结果准确无误，然后就应当把它保存起来。在保存文件时，只需选择 File 菜单中的 Save Project 命令即可。由于这是第一次保存文件，屏幕上会出现 Save File As 对话框，如图 4.8 所示。Form 的缺省名是 FORM1.FRM，为便于查找，可将其更名为 myfirst.frm，然后单击 OK 按钮即可。

接着，屏幕上又出现 Save Project As 对话框，如图 4.9 所示，用于保存 Project 文件。对话框中列出的缺省名为 PROJECT.MAK，可将其更名为 myfirst.mak，然后单击 OK 按钮即可。如果用户程序中有多个 Form 或模块，则每一个都出现一个对话框。如果是第二次存盘，这些对话框就不会出现了，因为 Visual Basic 已经知道在何处，使用什么样的文件名来保存文件。如果在第二次存盘时想在别的目录里，或者使用别的文件名保存文件，可以使用 File 菜单中的 Save File As 和 Save Project As 命令。


Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
1440, 3000
2415 x 495
Save File As
OK
Cancel
Network...
View Code
Form1
My Briefcase
Start
The Mi...
Project1
Microso...
myfile
Properti...
Save File As
7:48 PM

File Name: myfirst.frm

myfirst.frm
mysecond.frm

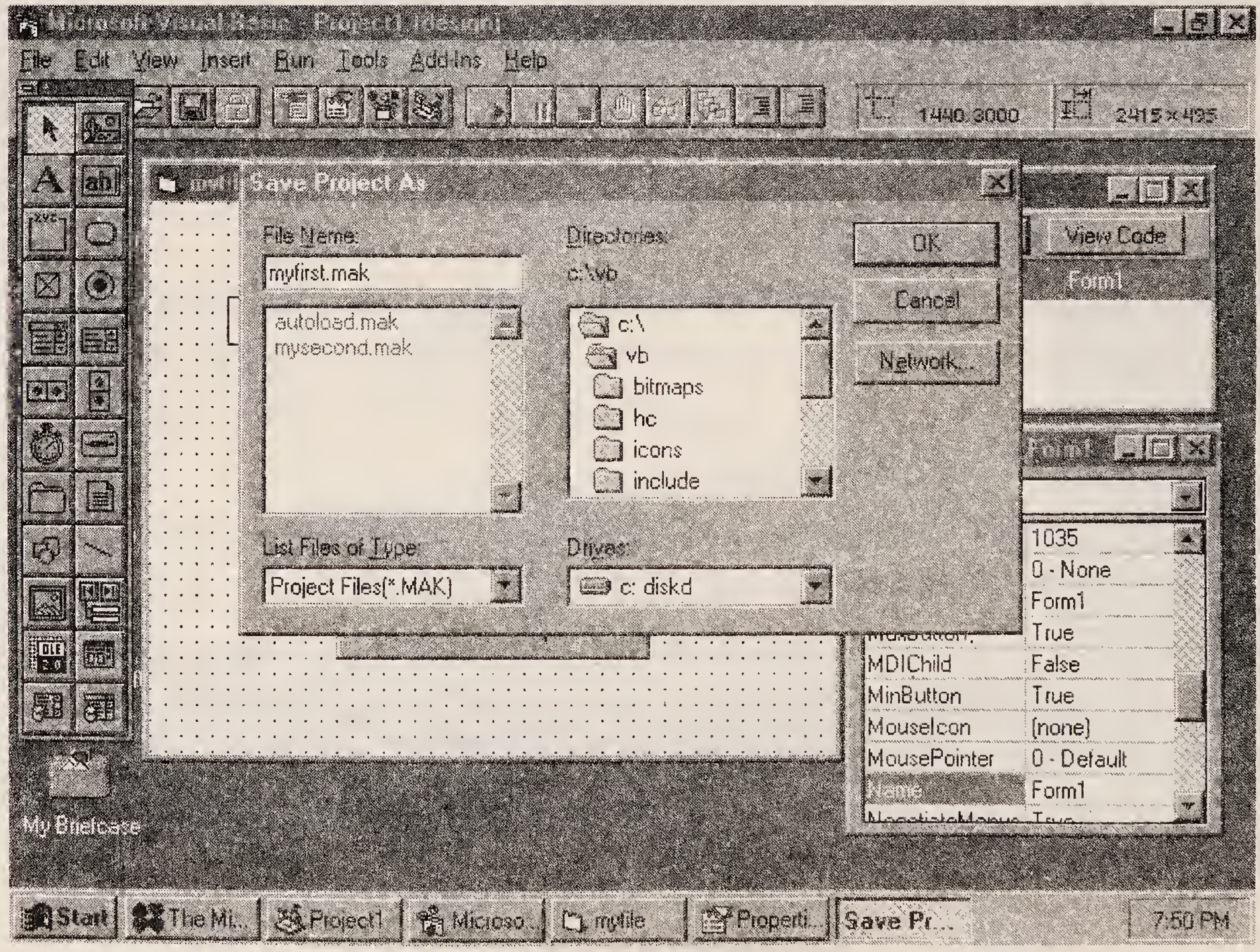
Directories: c:\vb

c:\
vb
bitmaps
hc
icons
include

List Files of Type: Form Files(*.FRM)

Drives: c: diskd

图 4.8 保存 File 对话框


Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
1440, 3000
2415 x 495
Save Project As
OK
Cancel
Network...
View Code
Form1
My Briefcase
Start
The Mi...
Project1
Microso...
myfile
Properti...
Save Pr...
7:50 PM

File Name: myfirst.mak

autoload.mak
mysecond.mak

Directories: c:\vb

c:\
vb
bitmaps
hc
icons
include

List Files of Type: Project Files(*.MAK)

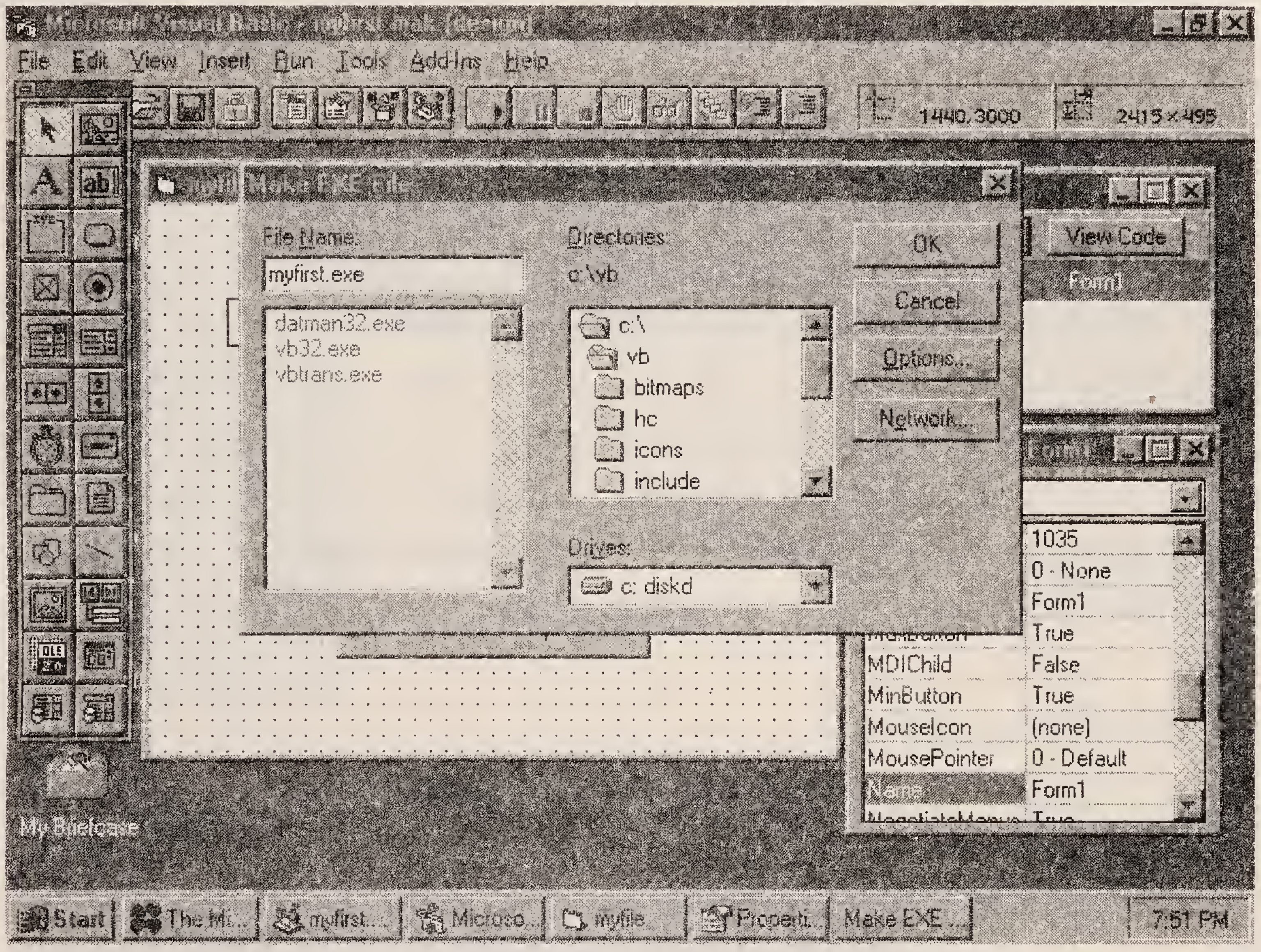
Drives: c: diskd

	1035
	0 - None
	Form1
	True
MDIChild	False
MinButton	True
MouseIcon	(none)
MousePointer	0 - Default
Name	Form1

图 4.9 保存 Project 对话框

4.5　编译应用程序

编译应用程序就是将程序编译成一个可独立运行的程序。如果程序此时在内存中，选择 File 菜单中的 Make EXE File 命令，则屏幕上会弹出一对话框，如图 4.10 所示。用户可以选择程序名以及所在的路径名。单击 OK 按钮后，程序就被编译并保存在磁盘上。但运行程序必须能访问到 Visual Basic 所包含的库文件 VBRUN200.DLL。


File
Edit
View
Insert
Run
Tools
Add-Ins
Help
1440, 3000
2415 × 495
Make EXE File
OK
Cancel
Options...
Network...
View Code
Form1
My Briefcase
Start
The Mi...
myfirst...
Microso...
myfile
Properti...
Make EXE ...
7:51 PM

File Name:
myfirst.exe
datman32.exe
vb32.exe
vbtrans.exe

Directories:
c:\vb
c:\
vb
bitmaps
hc
icons
include

Drives:
c: diskd

	1035
	0 - None
	Form1
	True
MDIChild	False
MinButton	True
MouseIcon	(none)
MousePointer	0 - Default
Name	Form1

图 4.10　Make EXE File 对话框

如需运行该程序，可关闭 Visual Basic，回到 Windows 95 桌面。然后单击 Start 进入 Programs 中，打开 VB 文件夹，在 VB 目录文件中，找到 MYFIRST.EXE 文件，并单击鼠标即可，也可以在 My Computer 中，进入 C 驱动器（VB 所在的驱动器），打开 VB 文件夹，找到 MYFIRST.EXE 文件图标，然后双击鼠标即可。

这就是生成、编译一个 Visual Basic 程序的整个过程，读者可以试着按这个程序操作一下，熟悉一下编程的步骤。

第五章　窗体和工具

在第三章的 Visual Basic 编程环境中，用户对窗体和工具有了简单的了解。本章将告诉用户怎样在给定条件下设计窗体，如何用工具箱向窗体添加控件，怎样利用属性窗口技术定制窗体和有关控件，如何填写代码使他们正常工作。最后讲到了有关 MDI 的基本概念。

5.1　基本概念

5.1.1　窗体

窗体是应用程序的编程窗口和对话框。窗体可包含有多个控件。用户程序可能需要除主程序窗口之外的一些其他不同的对话框，每个对话框或窗口都有自身的窗体。窗体就像一个可调查大小的绘制板，在绘制板上可绘制对象以组成一个可视的工程界面。程序运行时，用户希望看到的每一样东西都必须连接到窗体上。几乎每一个工程文件都至少与一个窗体相连。有时候一个应用程序含有多个窗体，为了避免混淆每个窗体得有不同的名字。窗体的 Name 属性允许给窗体设置合适的名字（有关窗体属性下节将详细介绍），但窗体命名必遵循以下规则：

（1）对象名必须以字母开头。

（2）对象名可包括字母、数字和下划线，而不能有空格和分号。

（3）最大字符数为 40。

（4）窗体名不能和像 Screen、Form 等对象重名。

5.1.2　控件

控件是所有用户加在窗体中用以使用户完成程序中相关任务的工具，常见的控件有命令按钮、文本框和复选框等。Visual Basic 4.0 标准工具箱提供了 24 种不同的控件。Visual Basic 4.0 标准版同时向用户提供了不同的定制控件，用户也可以从第三开发商手中购买不同的定制控件，这些新的定制控制使用户能够用 Visual Basic 设计出更加精彩的应用程序。控件是每个 Visual Basic 程序中的重要组成部分。后面相关章节，用户将会学习到更为详细的控件知识。

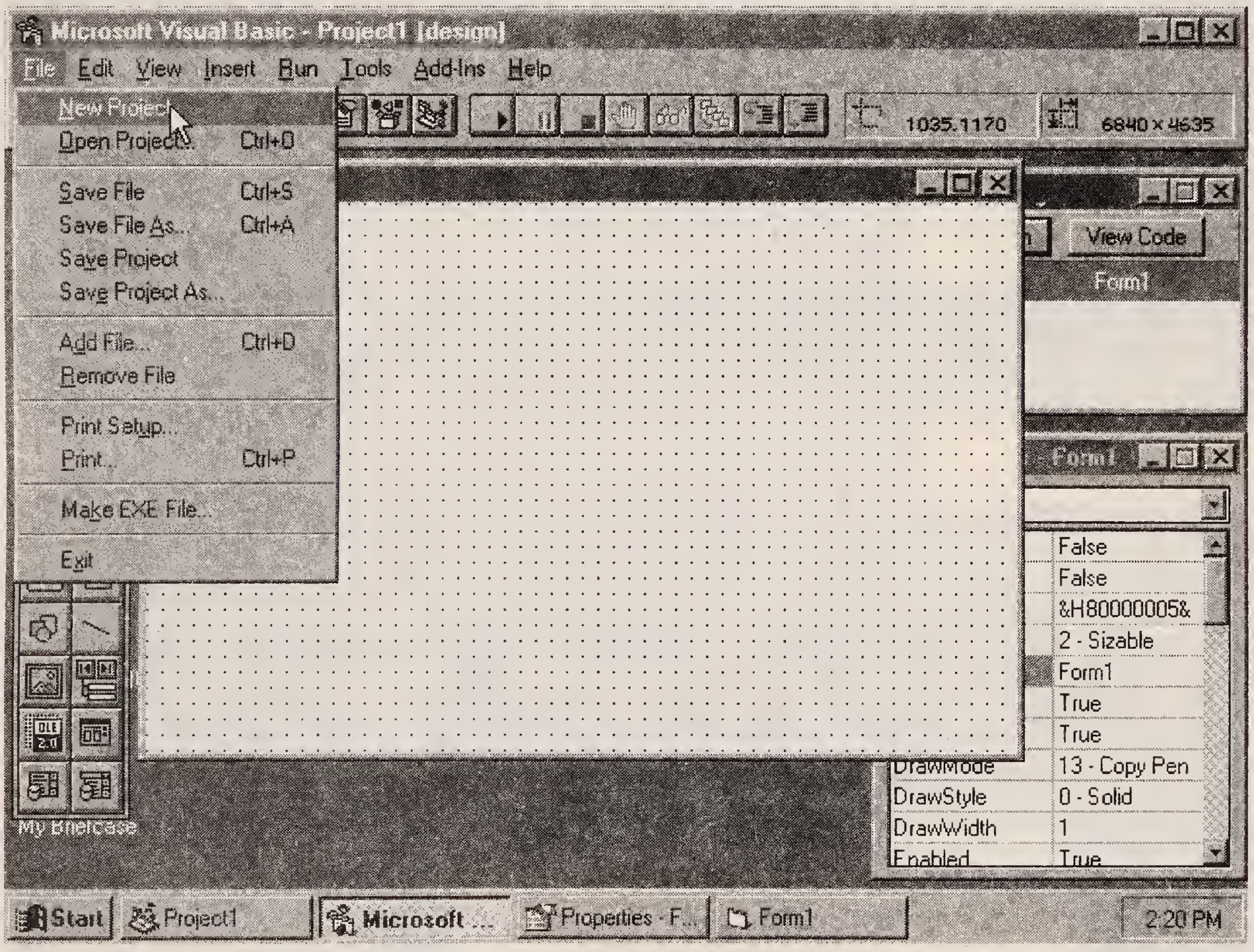
5.1.3　属性

前面已提到过窗体的 Name 属性，其实所有 Visual Basic 程序中对象都有属性，属性是 Visual Basic 程序中使用最频繁的概念之一。并且每个对象都包含有多个属性，对象包含的属性类型因对象不同而随之变化。例如：出现在文本框中文本为 Text 属性，窗体就没有 Text 属性，因为它不包含任何文本。

5.2　定制窗体

5.2.1　生成窗体

任何时候当用户想建立一个新的 Visual Basic 应用程序时都必须启动一个新的工程文件（工程文件是 Visual Basic 的技术术语，是 Visual Basic 对其开发的应用程序的称呼）。如果用户想这样做只需要单击 File 菜单，然后单击 New Project 命令或者键入“N”，见图 5.1。通常用户在启动 Visual Basic 时，Visual Basic 也会自动地给出一个缺省为 Form1 的工程文件。


Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
New Project
Open Project... Ctrl+O
Save File Ctrl+S
Save File As... Ctrl+A
Save Project
Save Project As...
Add File... Ctrl+D
Remove File
Print Setup...
Print... Ctrl+P
Make EXE File...
Exit
1035,1170
6840 x 4635
View Code
Form1
Form1
My Briefcase
Start
Project1
Microsoft ...
Properties - F...
Form1
2:20 PM

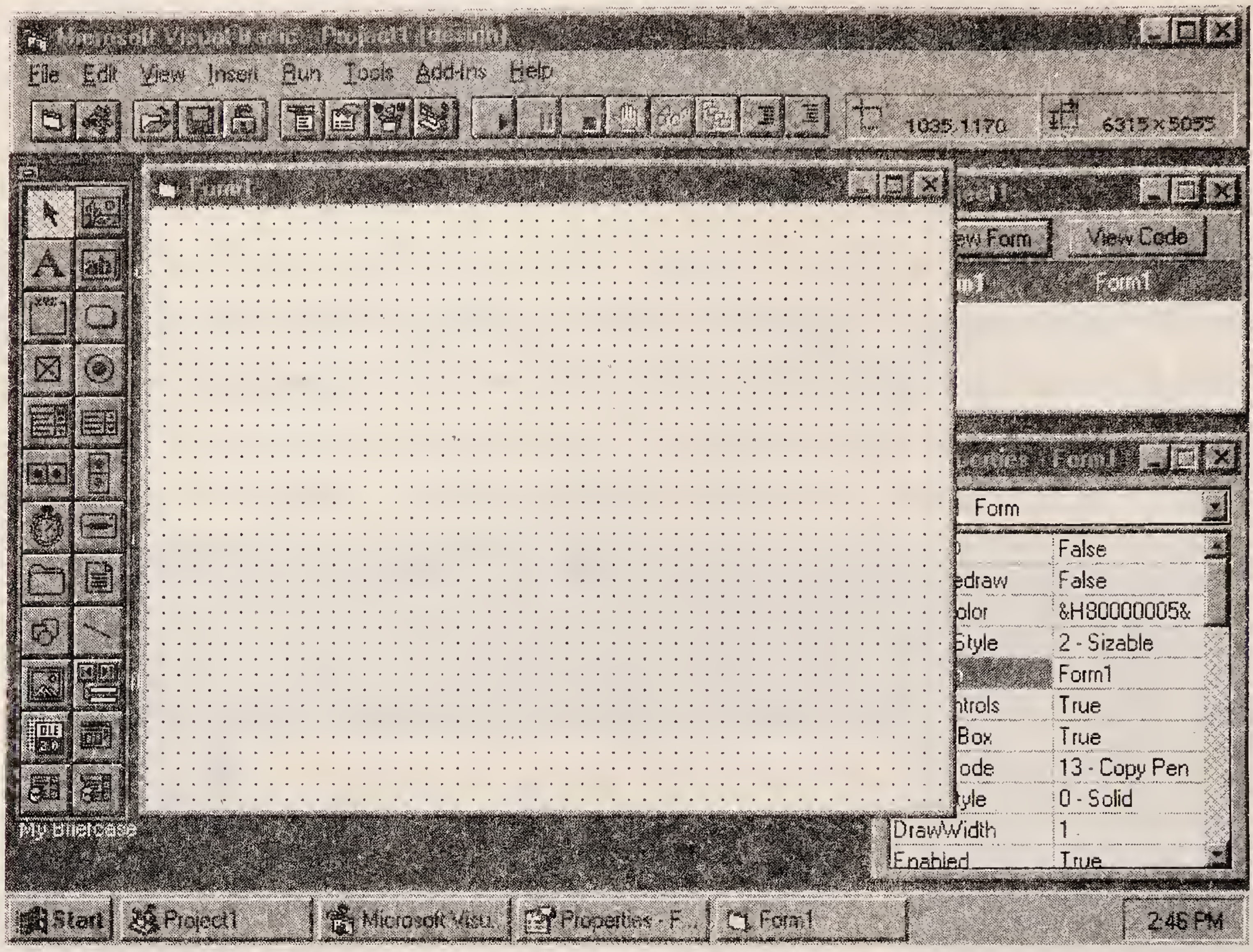
False
False
&H80000005&
2 - Sizable
Form1
True
True
DrawMode 13 - Copy Pen
DrawStyle 0 - Solid
DrawWidth 1
Enabled True

图 5.1　建立新工程文件

建立一个新的工程文件也常用工具条上的新工程文件图标，单击图标即可快捷地生成一个新的工程文件。

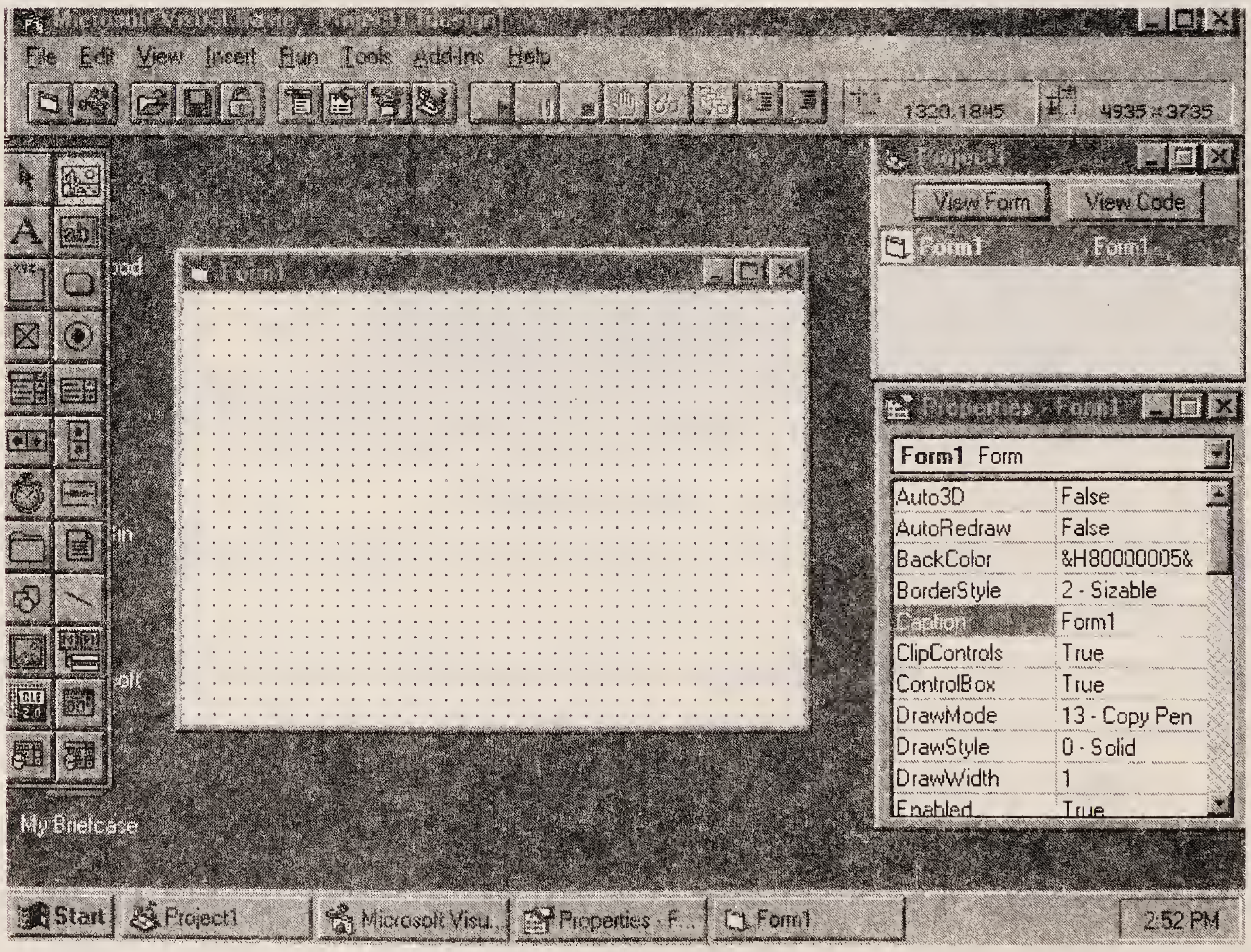
假定用户已启动了一个新的工程文件，这时屏幕变成图 5.2。其中屏幕中间有一个带网格状的点的窗口，这就是用户要定制的窗体。网格状点部分称为网格，它对精确地定位控件非常重要。空白窗体的顶部是题目条，里面有窗体的题目。

或许用户对窗体的大小和所处的位置均感到不满意，就像 Windows 95 中有些窗口一样，Visual Basic 允许用户任意改变窗体的大小和移动窗体的位置，并且他们的操作方法也是相同的。把鼠标指向窗体的边框，拖动鼠标就可修改窗体的大小。用户想移动窗体只需把鼠标指向题目条，然后拖动鼠标，即可移动窗体，具体操作可参考第一章 Windows 95 入门有关小节。图 5.3 就是窗体进行移动和改变大小后的显示结果。实际上这时用户已对窗体实现了初步定制。


Microsoft Visual Basic - Project1 [design]
File
Edit
View
Insert
Run
Tools
Add-Ins
Help
1035,1170
6315×5055
Form1
ew Form
View Code
m1
Form1
perties - Form1
Form
My Briefcase
Start
Project1
Microsoft Visu...
Properties - F...
Form1
2:46 PM

D	False
edraw	False
olor	&H80000005&
Style	2 - Sizable
n	Form1
ntrols	True
Box	True
ode	13 - Copy Pen
tyle	0 - Solid
DrawWidth	1
Enabled	True

图 5.2　打开的屏幕


Microsoft Visual Basic - Project1 [design]
File
Edit
View
Insert
Run
Tools
Add-Ins
Help
1320,1845
4935×3735
Project1
View Form
View Code
Form1
Form1
Form1
Properties - Form1
Form1 Form
My Briefcase
Start
Project1
Microsoft Visu...
Properties - F...
Form1
2:52 PM

Auto3D	False
AutoRedraw	False
BackColor	&H80000005&
BorderStyle	2 - Sizable
Caption	Form1
ClipControls	True
ControlBox	True
DrawMode	13 - Copy Pen
DrawStyle	0 - Solid
DrawWidth	1
Enabled	True

图 5.3　窗体的初步定制

通过比较图 5.2 和图 5.3，用户可以看到通过对窗体的定制屏幕更为清晰明朗，便于用户操作。此时窗体的尺寸和位置就是用户在运行程序时所看到的尺寸和位置，当然 Visual Basic 也允许用户在运行时重新定制窗体，这将在后面有关章节讨论。

5.2.2　窗体的几种快捷操作

有时用户想把主要精力放在窗体中，为了不受其他窗口的影响希望把窗体扩充到整个屏幕，这时用户只需单击位于窗体右上端第二个的最大化图标。有时用户在代码窗口工作，为了避免过多的窗口重叠影响用户视觉可将窗体最小化在任务栏上或者把窗体隐藏起来。用户只需单击窗体右上端第一个最小化图标或者第三个隐藏图标。如果用户想恢复最小化只需单击任务栏上相应的按钮。显示被隐藏的窗体可打开 View 菜单单击 Form 命令，或键入 F，见图 5.4，也可使用快捷键 Shift＋F7。

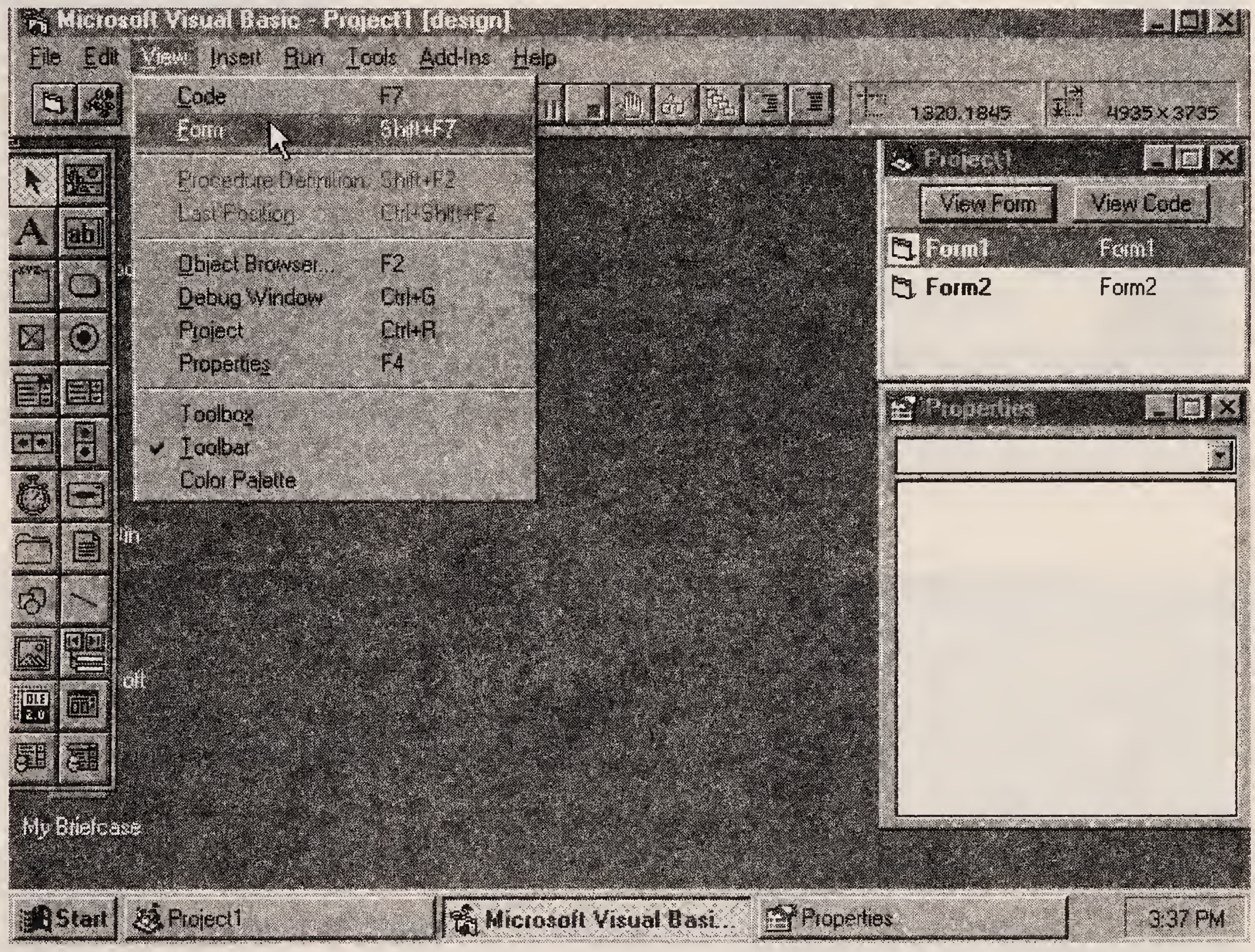

Microsoft Visual Basic - Project1 [design]
File
Edit
View
Insert
Run
Tools
Add-Ins
Help
Code　F7
Form　Shift+F7
Procedure Definition　Shift+F2
Last Position　Ctrl+Shift+F2
Object Browser...　F2
Debug Window　Ctrl+G
Project　Ctrl+R
Properties　F4
Toolbox
✓ Toolbar
Color Palette
1320,1845
4935×3735
Project1
View Form
View Code
Form1　Form1
Form2　Form2
Properties
My Briefcase
Start
Project1
Microsoft Visual Basi...
Properties
3:37 PM

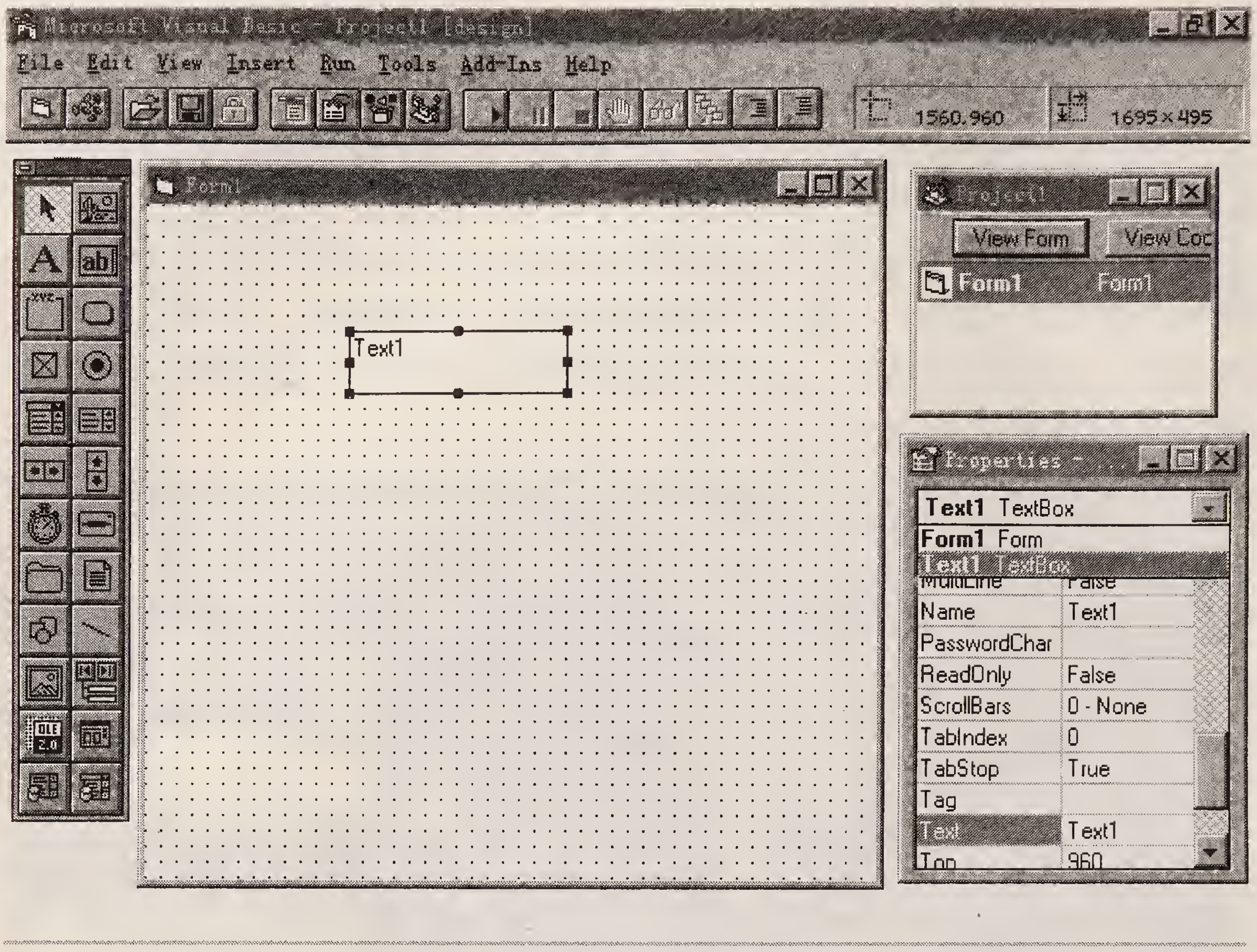
图 5.4　显示被隐藏的窗体

5.2.3　属性＜Properties＞窗口

用户只需单击任务栏中属性窗口按钮就由窗体切换到属性窗口，使属性窗口被激活。如果屏幕上没有显示属性窗口，用户只需键入 F4 显示属性窗口。用户也可以单击工具条上的快捷键图标显示属性窗口。

属性窗口包含被选择的窗体、控件在设计时拥有的属性的列表。这些属性能够在设计时被改变，当用户选择多个控件时，属性窗口包含的属性列表被所选择的控件共享。

在属性窗口的顶部是题目条，在题目条下面是对象框。对象框中显示当前工程文件被选择的对象——控件或窗体，单击对象框右边的箭头，用户可以看到工程文件中被选择的对象列表，见图 5.5。在这个列表中，用户可选择想改变它的属性的窗体和控件。


Microsoft Visual Basic - Project1 [design]
File
Edit
View
Insert
Run
Tools
Add-Ins
Help
1560,960
1695×495
Form1
Text1
Project1
View Form
View Coc
Form1　Form1
Properties - ...

Text1 TextBox

Form1 Form

Text1 TextBox

MultiLine	False
Name	Text1
PasswordChar	
ReadOnly	False
ScrollBars	0 - None
TabIndex	0
TabStop	True
Tag	
Text	Text1
Top	960

图 5.5　属性窗口和对象列表

在图 5.5 中，对象列表中只有两个对象 Form1 和 Text1，它们代表对象名（本例是缺省名），用户可以对它们修改。Textbox 和 Form 代表对象类型，此处是文本框和窗体。设置对象属性的一般步骤为：

(1) 单击对象框右边的箭头打开对象列表，再单击用户需要设置其属性的对象见图 5.5。

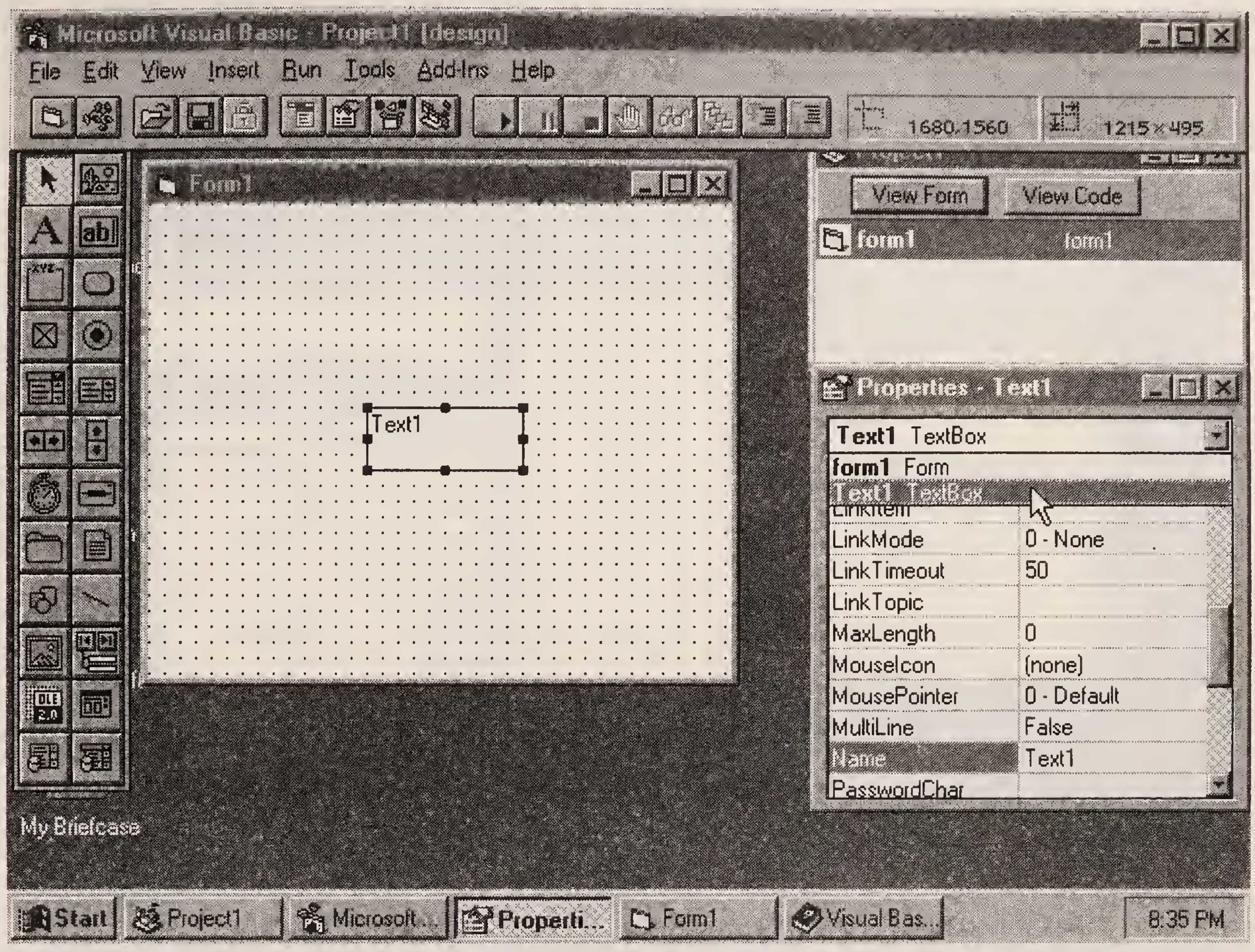
(2) 在属性列表中单击用户想设置属性的属性名。这一工作也可通过键入 Ctrl+Shift+属性名的第一个字母来完成，如果有几个属性均以相同的字母开始，继续键入 Ctrl+Shift+第一个字母直到用户想选择的属性被选择。

Visual Basic 在设置框中显示被选择属性的当前设置。

(3) 在设置框中，键入用户的设置。

例如，用户想设置文本框的 Name 属性，打开对象列表框单击“TextBox”见图 5.6，属性列表中显示出与文本框相关的一些属性和这些属性的当前设置。然后单击 Name 属性（如果用户没有发现 Name 属性，可以单击上、下箭头来滚动列表框来寻找），见图 5.7。此时用户可看到文本框当前 Name 属性为 Text1，键入 input text 回车就把文本框的 Name 设为 input text，这时对象框中名字随之发生变化见图 5.8。

在属性列表中，有的属性的设置情况范围是有限的。对于这类属性当用户单击它时，在设置框的右边有一个向下箭头。用户单击箭头打开可供用户选择的设置，单击列表中的某一项设置就可完成属性设置，见图 5.9。


Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
1680,1560
1215 x 495
Form1
Text1
View Form
View Code
form1
form1
Properties - Text1
My Briefcase
Start
Project1
Microsoft ...
Properti...
Form1
Visual Bas...
8:35 PM

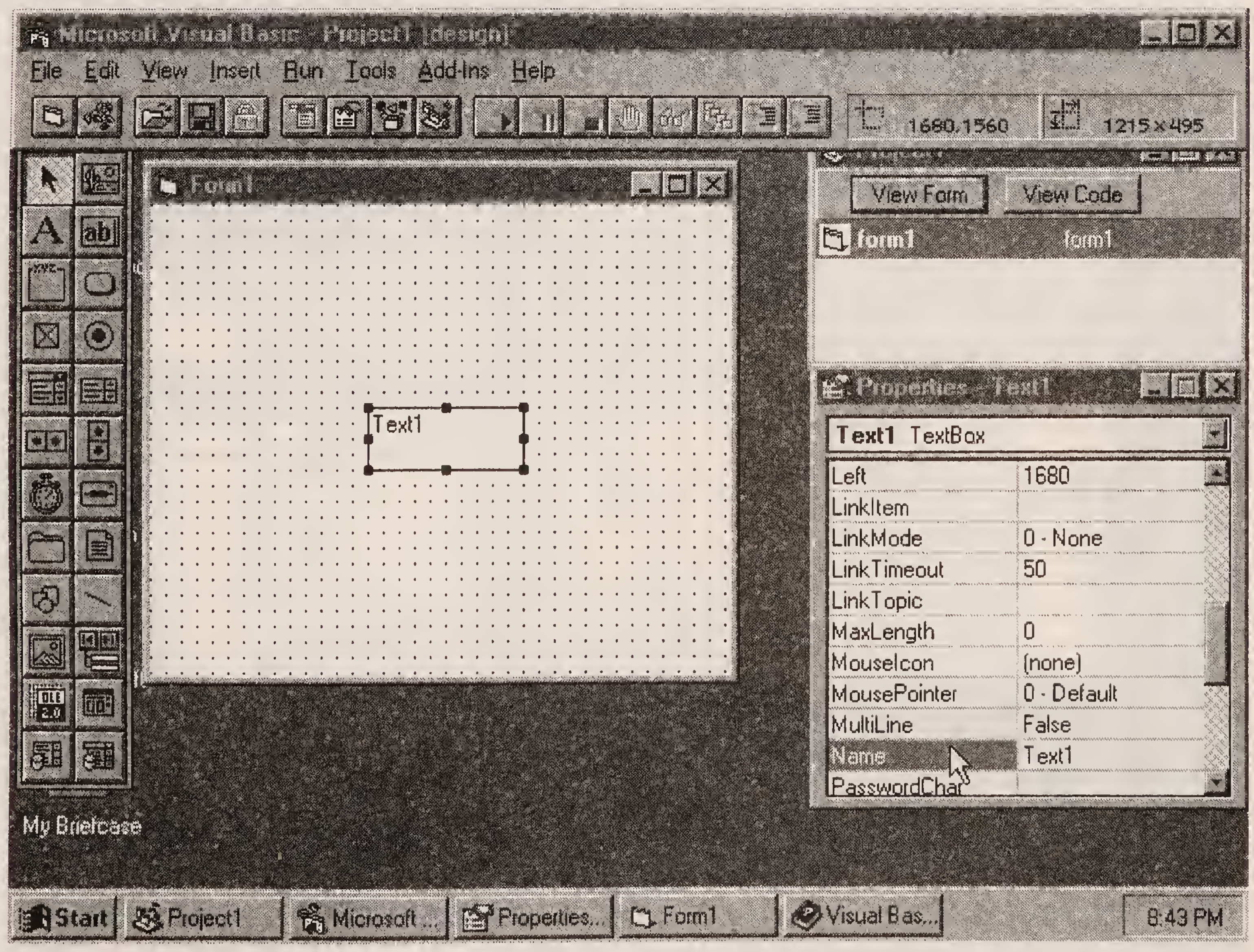
Text1 TextBox

form1 Form

Text1 TextBox

LinkItem	
LinkMode	0 - None
LinkTimeout	50
LinkTopic	
MaxLength	0
MouseIcon	(none)
MousePointer	0 - Default
MultiLine	False
Name	Text1
PasswordChar	

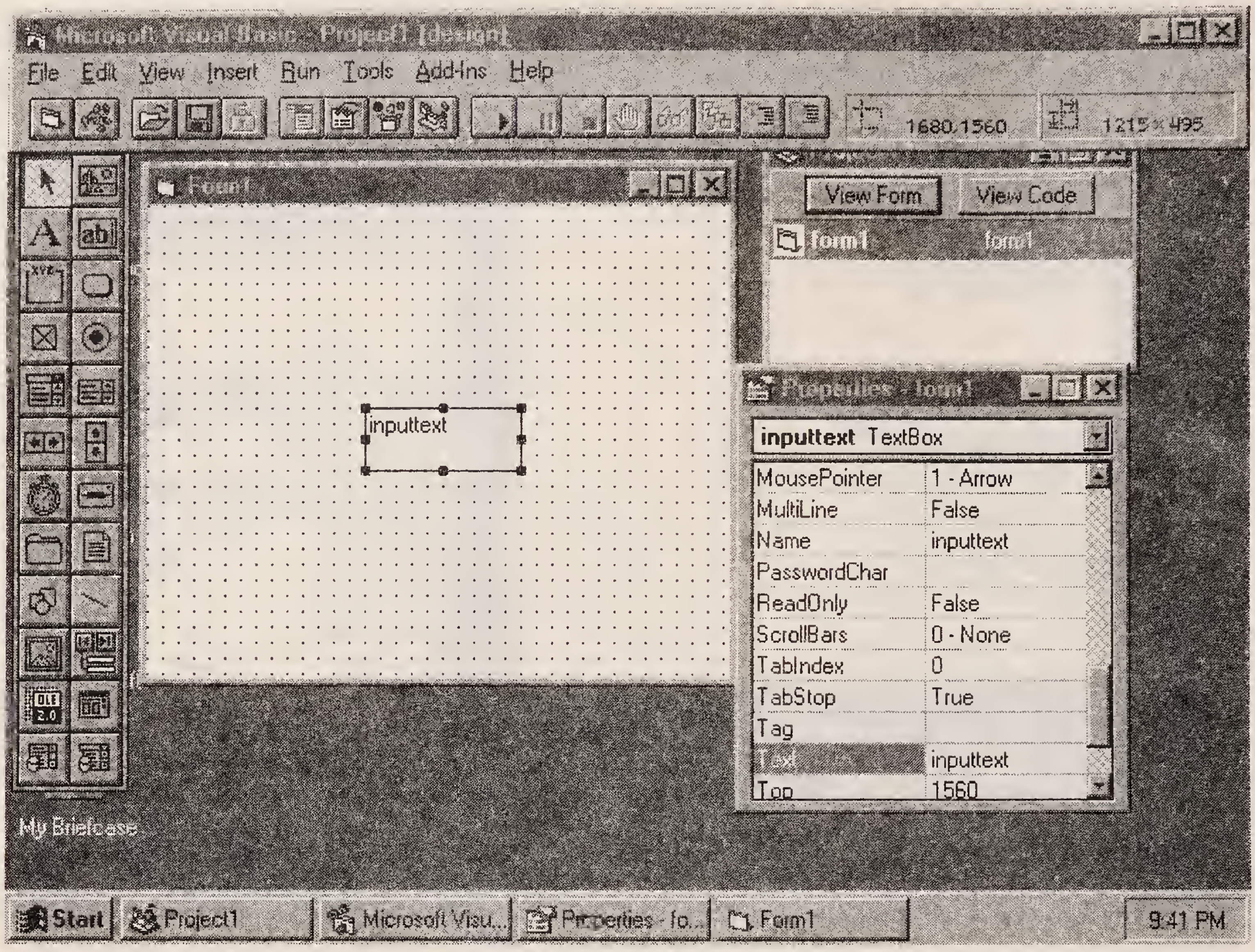
图 5.6　对象的选择


Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
1680,1560
1215 x 495
Form1
Text1
View Form
View Code
form1
form1
Properties - Text1
My Briefcase
Start
Project1
Microsoft ...
Properties...
Form1
Visual Bas...
8:43 PM

Text1 TextBox

Left	1680
LinkItem	
LinkMode	0 - None
LinkTimeout	50
LinkTopic	
MaxLength	0
MouseIcon	(none)
MousePointer	0 - Default
MultiLine	False
Name	Text1
PasswordChar	

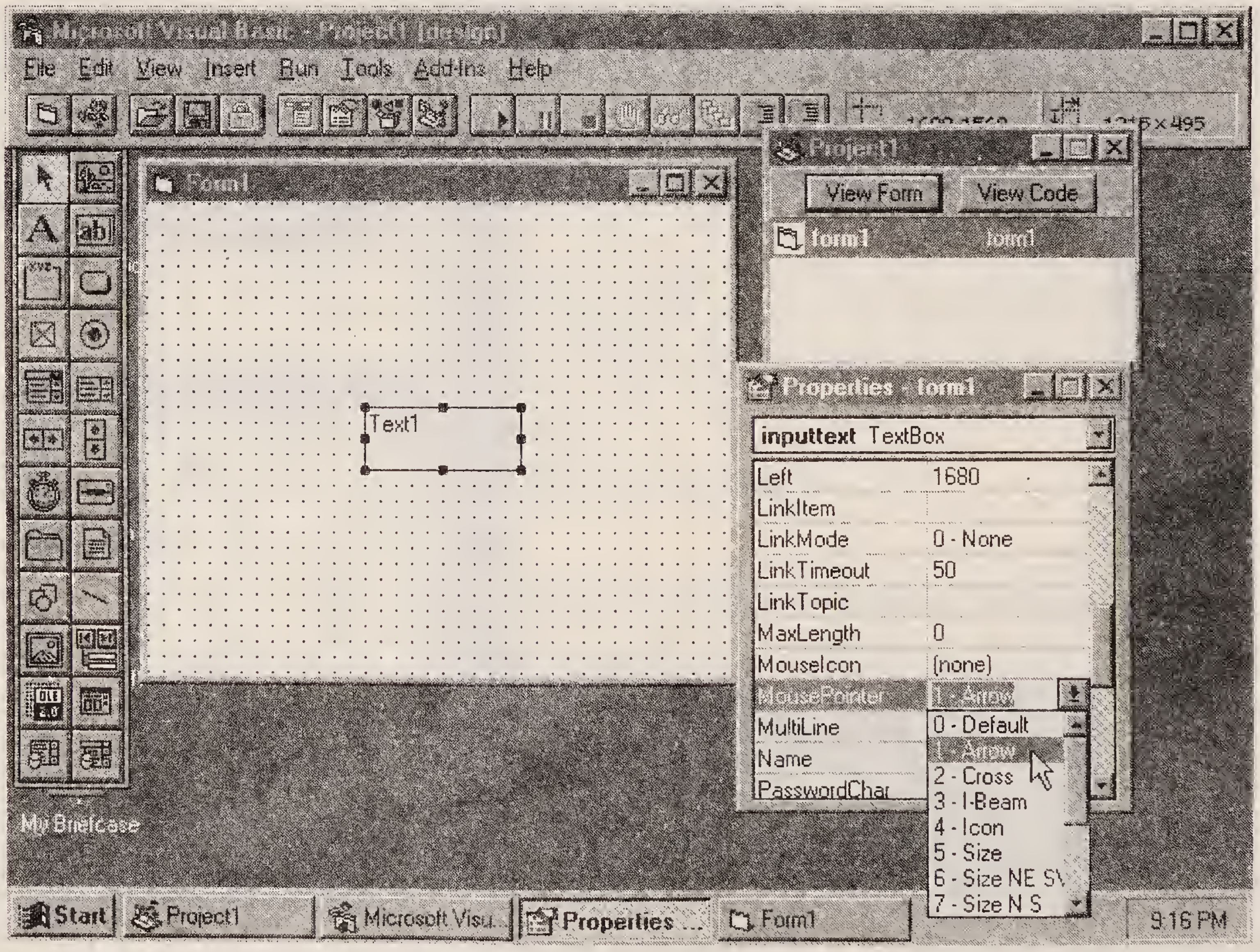
图 5.7　Name 属性的选择


Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
1680,1560
1215 x 495
Form1
inputtext
View Form
View Code
form1 form1
Properties - form1
My Briefcase
Start
Project1
Microsoft Visu...
Properties - fo...
Form1
9:41 PM

inputtext TextBox

MousePointer	1 - Arrow
MultiLine	False
Name	inputtext
PasswordChar	
ReadOnly	False
ScrollBars	0 - None
TabIndex	0
TabStop	True
Tag	
Text	inputtext
Top	1560

图 5.8 Text 属性的设置


Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
1215 x 495
Form1
Text1
Project1
View Form
View Code
form1 form1
Properties - form1
My Briefcase
Start
Project1
Microsoft Visu...
Properties ...
Form1
9:16 PM

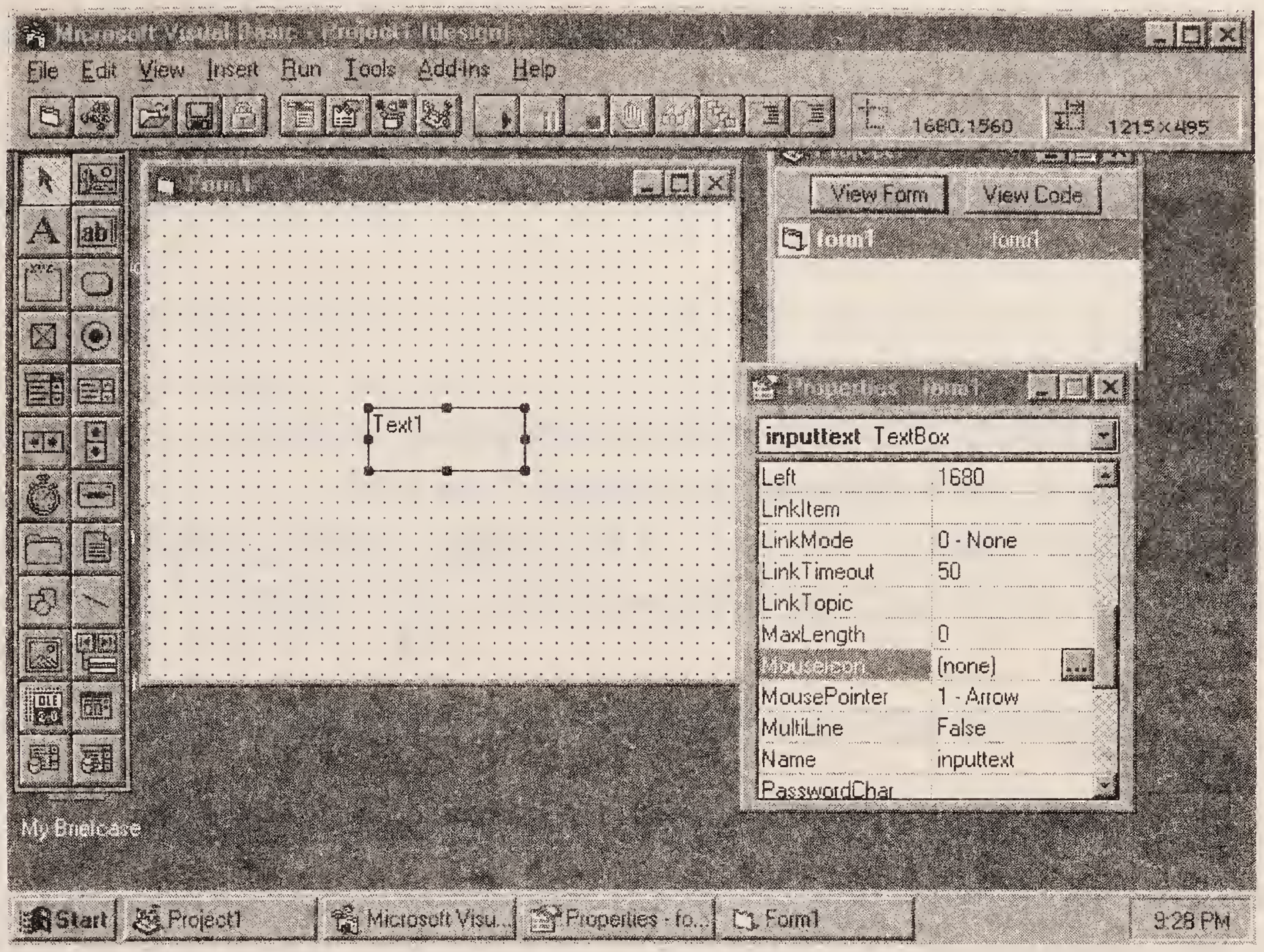
inputtext TextBox

Left	1680
LinkItem	
LinkMode	0 - None
LinkTimeout	50
LinkTopic	
MaxLength	0
MouseIcon	(none)
MousePointer	1 - Arrow
MultiLine	
Name	
PasswordChar	

0 - Default
1 - Arrow
2 - Cross
3 - I-Beam
4 - Icon
5 - Size
6 - Size NE SW
7 - Size N S

图 5.9 有限种设置情况的属性选择

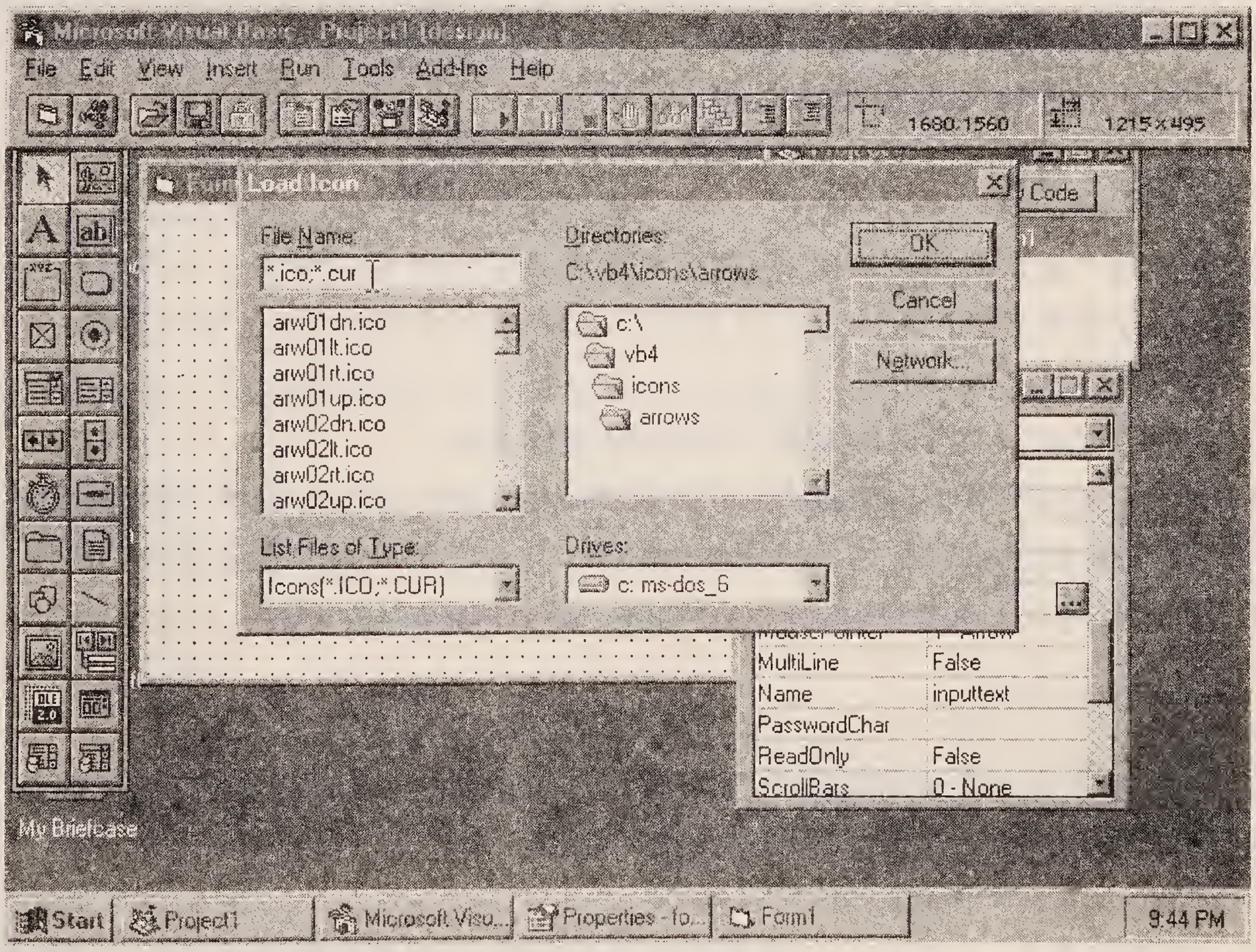
在属性列表中，当用户选择某项属性时，设置框中出现了省略号按钮，单击此按钮出现一对话框，用户可利用对话框设置相关属性，见图 5.10 和图 5.11。


Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
1680,1560
1215×495
Form1
Text1
View Form
View Code
form1
Properties - form1
My Briefcase
Start
Project1
Microsoft Visu...
Properties - fo...
Form1
9:28 PM

inputtext TextBox

Left	1680
LinkItem	
LinkMode	0 - None
LinkTimeout	50
LinkTopic	
MaxLength	0
MouseIcon	(none)
MousePointer	1 - Arrow
MultiLine	False
Name	inputtext
PasswordChar	

图 5.10　带有省略号按钮的属性


Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
1680,1560
1215×495
Load Icon
OK
Cancel
Network...
arw01dn.ico
arw01lt.ico
arw01rt.ico
arw01up.ico
arw02dn.ico
arw02lt.ico
arw02rt.ico
arw02up.ico
c:\
vb4
icons
arrows
Code
My Briefcase
Start
Project1
Microsoft Visu...
Properties - fo...
Form1
9:44 PM

File Name: *.ico;*.cur

Directories: C:\vb4\icons\arrows

List Files of Type: Icons(*.ICO;*.CUR)

Drives: c: ms-dos_6

MultiLine	False
Name	inputtext
PasswordChar	
ReadOnly	False
ScrollBars	0 - None

图 5.11　设置对话框

有时用户想设置一属性为工程文件中某几个控件所共享，用户必须选择这些控件，然后按上述步骤设置属性，这一设置将影响被选择的所有控件。

5.2.4 窗体的属性

前面用户已接触到了窗体的 Name 属性，其实窗体的属性很多。例如 BorderStyle、Caption、ControlBox、DrawStyle 属性等，这些属性在后面的应用程序设计和开发中经常要用到，下面将对窗体的常用属性逐个详细介绍。

1. Name 属性

在编写代码时该属性为窗体指定一个名字，此属性在运行时无效。

语法：object . Name（此处 object 代表窗体 Form，下同）

一新窗体的缺省名是窗体 Form 加上一个特定的整数，例如，第一个新窗体是 Form1。

注意　窗体不能用 Visual Basic 中的关键字来命名，这样可能在用户的代码中引起冲突。

2. Caption 属性

此属性确定窗体的标题条中所显示的文本，当窗体被最小化时，该文本被显示在窗体图标的下面。

语法：object. Caption[＝String]

当用户创建一个新窗体时，其标题条中的缺省文本也是窗体 Form 加上一个特定的整数，例如 Form1 等。要显示标题，需把 BorderStyle 属性设置为 1 或 2 或 3(见下述)。若标题对窗体的标题条来说太长，则标题会被剪去一部分。

3. ControlBox 属性

该属性当用户运行应用程序时有效，用来在窗体题目条左边设置一个控制框，单击控制框显示出一个任务列表，如缩小、放大和关闭等，如果存在热键的话，还列出热键。

语法：object. ControlBox

其中 object 代表一个有效的对象，在此处是窗体 Form。

ControlBox 属性设置情况有两种：True 和 False。True 是缺省设置，能够使窗体显示一个控制菜单。False 设置使控制菜单从窗体上消失。为了在题目条的左边显示控制框，BorderStyle 属性必须设置为 1(Fixed Single)、2(Sizable)或 3(Fixed Double)。

4. BorderStyle 属性

该属性为窗体等对象设置边界类型，用户可以为该属性设置四种值，缺省值为 2，允许用户通过窗体边缘上的热点改变窗体的大小和形状。

语法：object. BorderStyle＝[Value]

Value 值可以是 0、1、2 或 3。0 设置使窗体无框架，1 设置使窗体大小不变(在设计时被指定)且具有单线框架，2 设置使窗体的大小可改变且具有双线框架，3 设置使窗体的大小不变(在设计时被指定且具有双线框架)。窗体的 BorderStyle 属性决定一个可显示窗体的关键

特征。

注意　如果一个带有菜单的窗体的 BorderStyle 属性被设置为 3(Fixed Double)，它用双固定边框显示框体。

5. AutoRedraw 属性

该属性控制窗体图像的重建。

语法:object . AutoRedraw[=boolean]

boolean 可被设置两种情况:True 和 False。在其他窗口覆盖某窗体后，又返回该窗体时，如果将 AutoRedraw 设置为 True，Visual Basic 将自动刷新或者重画该窗体的所有图形。如果将该属性设置为 False，Visual Basic 必须调用一事件过程来执行该项任务。此特性是使用图形方法包括 Circle、Cls、Point 和 Print 的核心(详见图形操作一章)，设置 AutoRedraw 为 True，可以在窗体中重画这些方法的输出。

6. BackColor 和 ForeColor 属性

BackColor 属性决定窗体的背景颜色，ForeColor 属性决定窗体的前景颜色。

语法:object. BackColor[=Color]
object. ForeColor[=Color]

Visual Basic 利用 Windows 操作环境 RGB 颜色方案设置对象颜色，每个特性有以下范围内的设置:正常 RGB 颜色、系统缺省颜色，其中正常 RGB 颜色设置使用颜色调色板或在代码中使用 RGB 或 QBColor 函数确定，系统缺省颜色设置是用 Visual Basic 的 Constant. TXT 文件中系统缺省颜色常量指定颜色，对于所有的窗体，设计时的缺省值如下:

BackColor ——设置成 WINDOW _ BACKGROUND 的系统缺省颜色。

ForeColor ——设置成 WINDOW _ TXT 的系统颜色。

以上两种值均在 Constant. TXT 中被声明。

7. Icon 属性

该属性决定在运行时窗体被最小化时所显示的图标。

语法:object. Icon

用户能够安排一个唯一的图标给窗体用来表示窗体的功能。用户可以在设计时用属性窗口为窗体设置图标，如果用户没有给窗体指定图标，Visual Basic 会为窗体设置一个缺省图标。用户在为窗体等对象设置图标时，可利用 Visual Basic 的图标库，该图标库在 ICONS 子目录下。

注意　用户为了能够看到窗体的图标，窗体必须被最小化，BorderStyle 属性必须被设置为 1(Fixed Single)或 2(Sizable)，MinButton 属性必须被设置为 True。

8. MaxButton 属性

该属性决定在运行时窗体是否有最大化按钮。

语法:object . MaxButton

MaxButton 属性有两种设置:True 和 False。True 是缺省设置，表明窗体有最大化按钮，

False 表明窗体没有最大化按钮。最大化按钮能够让用户把窗体扩大到整个屏幕大小，为了能够显示最大化按钮，BorderStyle 属性应被设置为 1(Fixed Single)或 2(Sizable)。当一个窗体被最小化时，最大化按钮会自动变为恢复按钮。

9. MinButton 属性

该属性决定窗体在运行时是否有最小化按钮。

语法：object. MinButton

MinButton 属性也有两个设置值：True 和 False，缺省设置是 True，它表明窗体有最小化按钮，False 表明窗体没有最小化按钮。同 MaxButton 属性一样，为了能够显示最小化按钮，BorderStyle 属性应该被设置为 1 或 2。

注意 WindowState 属性影响当前窗体的状态，如果用户把 WindowState 属性设置为 2(Maximized)，或 1(Minimized)，无论 MaxButton 或 MinButton，BorderStyle 属性被设置成什么，窗体都被最大化或最小化。

10. WindowState 属性

该属性决定窗体看起来像什么样子，有三种设置：0、1 或 2。0 是缺省设置，1 是把窗体最小化成图标，2 是放大窗体。

语法：object. WindowState[=Value]

Value 可以是 0、1 或 2。

11. Enabled 属性

该属性决定窗体是否对用户产生的事件发生反应。

语法：object. Enabled[=boolean]

该属性有两种设置：True 和 Flase，True 允许窗体对事件做出反应，False 禁止窗体对事件做出反应。

12. FontBold、FontItalic、FontStrikethru，FontUnderline 和 FontTransparent 属性

前四种属性确定下列四种字形：黑体，斜体、删除体和下划线体，FontTransparent 属性决定窗体中背景文字或图形是否包括在特殊字体的字符中。

语法：object. FontBold[=boolean]
object. Fontltalic[=boolean]
object. FontStrikethru[=boolean]
object. FontUnderline[=boolean]
Transparent[=boolean]

这些特性的设置情况有两种：True 和 False。True 设置值按缺省的 FontBold 和 Font Transparent 把字型格式化，允许背景文字或图形通过字符字体显示。False 设置值使用这些字体格式化文本。不管是在设计时使用窗口特性，还是在运行时使用代码设置，设置这些特性不会影响已经画到窗体上的图形或打印输出。

13. Fontsize 属性

确定窗体中文本所用字体的大小。

语法:object. Fontsize[=points]

用该特性去指定文本所需字体的大小,缺省时由系统定义。用户要想改变缺省值,可用点指定字体的大小。FontSize 最大的数值是 2048 点。

14. Picture 属性

该属性确定窗体中一个图片被显示。

语法:object. Picture[=picture]

该属性的设置值有:none、Bitmap、icon、metafile。none 设置值是缺省设值,用来确定窗体中没有图片。Bitmap、icon、metafile 等设置值用来指定一个图形。如果在设计时设置属性,可以从 Properties 窗口选择它。如果在运行时设置该属性,则必须使用 Load Picture 函数。

15. Height,Width 属性

该属性确定窗体外部的高度和宽度,包括边框和标题条。

语法:object. Width[=number]
object. Height[=number]

对于窗体,该属性总是用点素(twips)来测量。对于一个窗体,这些属性随用户或代码确定的对象尺寸而改变,这些属性的最大值随系统而定。

16. Left,Top 属性

Left 属性确定窗体内部最左端与它的包容器最左端之间的距离,Top 属性确定窗体内部最上端与它的包容器最上端之间的距离。

语法:object. Left=[x]
object. Top=[y]

通常,Left 和 Top 属性在一个窗体中总是成对出现的,当通过用户或通过代码移动窗体时,这两个属性的值随之改动。

17. Visible 属性

该属性设置一个值确定窗体是被显示还是被隐藏。

语法:object . Visible[=boolean]

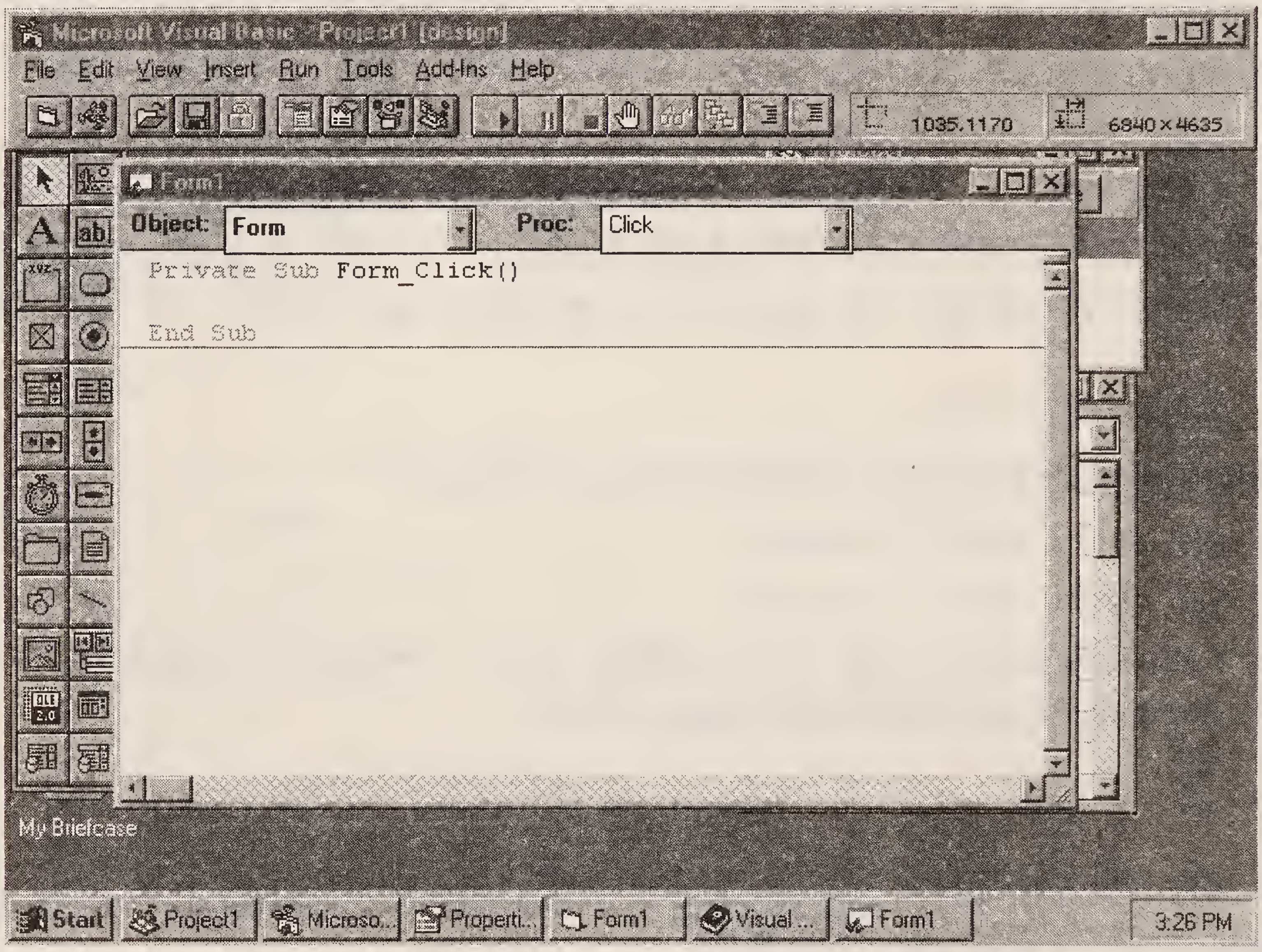
该属性有两种设置情况:True 和 False,True 是缺省设置,它能够使窗体可见。如果用户将该属性设置为 False 窗体将被隐藏。用户若要在启动时隐藏一个对象,可在设计时设置 Visible 属性为 False。在代码设置该属性使对象被隐藏之后,在运行时由某一事件来激活并显示对象。

注意　将 Show 或 Hide 方法应用于一个窗体,这相当于在程序中将 Visible 属性设置为 True 或 False。

5.2.5　窗体的相关事件

有关事件的概念在 Visual Basic 概述中有所叙述，相信用户对它已有了一个了解，在此不再多讲。与窗体相关的重要事件有以下几种：

(1) Click 事件：当用户在窗体上按一下鼠标按钮（单击）时，Visual Basic 将调用 Form_Click 过程，见图 5.12。如果用户在窗体所含有的对象上单击一下鼠标，而传给被单击的对象，即使该对象未处理该事件，Click 事件将不会传给窗体。


Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
1035,1170
6840 x 4635
Form1
Object: Form
Proc: Click
My Briefcase
Start
Project1
Microso...
Properti...
Form1
Visual ...
Form1
3:26 PM

```
Private Sub Form_Click()

End Sub
```

图 5.12　调用事件过程

(2) DblClick 事件：用户在窗体上的同一点连续地按两下鼠标按钮（双击）时，窗体接受该事件过程。

(3) Load 事件：一旦装载窗体，如启动应用程序，则自动产生该事件。Load 事件很适用于在启动程序时对属性和变量的初始化。

(4) GotFocus，LostFocus 事件：当窗体收到或失去焦点时，无论用户如何动作，GotFocus 或 LostFocus 事件都会发生。

(5) Unload 事件：当打算把一个窗体从屏幕上删除时发生，当该窗体再被装载时，它的所有控件都要重新初始化。这个事件是由用户动作（用控件菜单关闭窗体）或一个 Unload 语句触发的。

(6) Activate，Deactivate 事件：当窗体变成活动窗口时发生 Activate 事件，在另一个窗体变成活动窗口前发生 Deactivate 事件。窗体可通过用户的操作变成活动窗体，如鼠标单击窗体或控件，或在代码中使用 Show 或 SetFocus 方法。

(7) Paint 事件：窗体被移动或放大时，或者一个覆盖对象的窗口移动后，窗体暴露出来，会发生此事件。

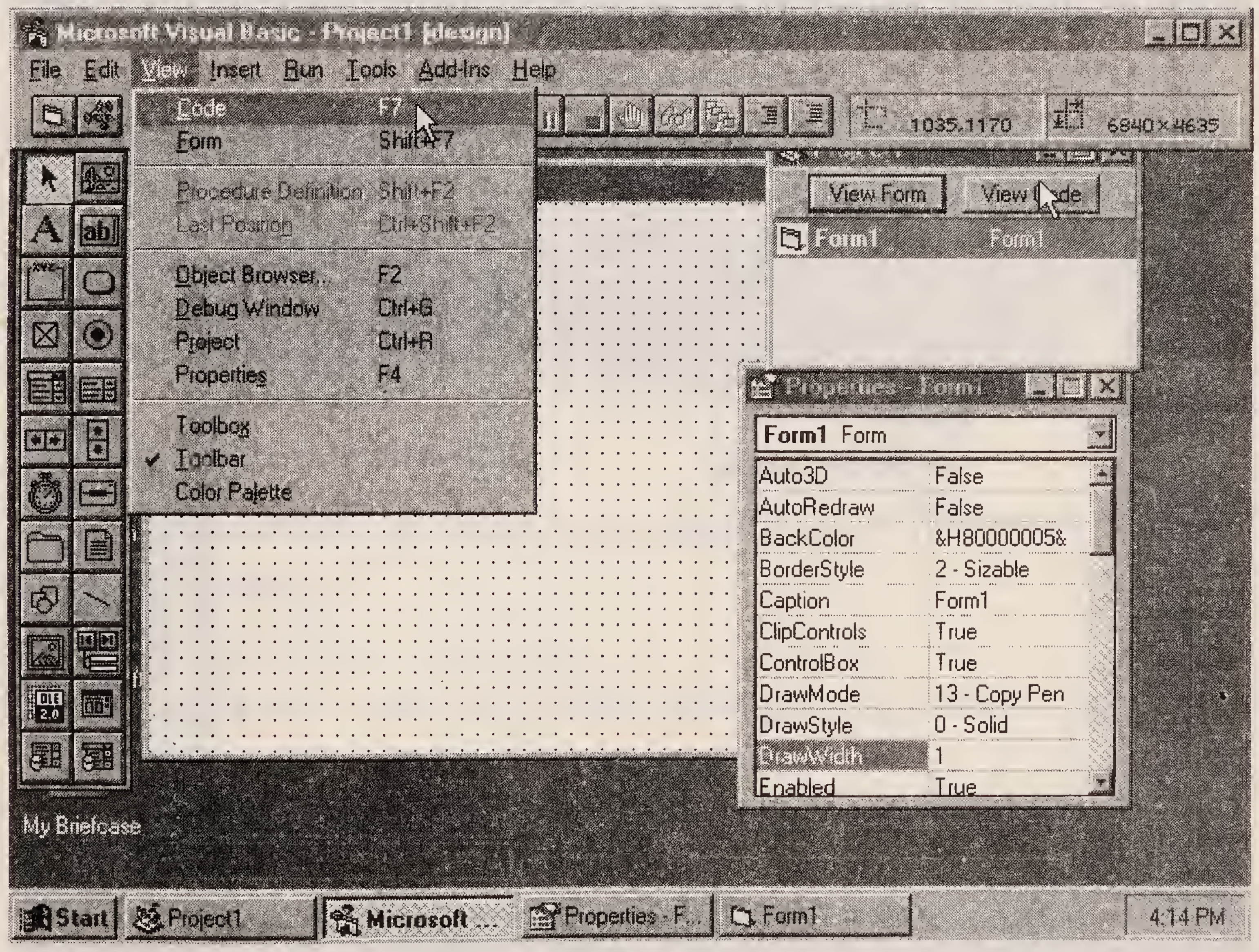
同窗体相关的事件还很多，例如 DragDrop、DragOver、KeyPress、KeyDown、Resize、MouseUp、MouseDown 等，以上几种只是最常用的几种，其他的相关事件用户可参考 Visual Basic 4.0 参考手册。

5.2.6　窗体对事件的响应

窗体能够识别的事件很多，但是如果用户不编写代码告诉它们当事件发生该怎样做，则窗体将不会采取任何响应，用户必须在事件过程中编写代码告诉 Visual Basic 怎样做，下面用户将学如何编写简单代码让窗体对单击（Click）、双击（DblClick）、装载（Load）事件发生响应。或许用户是初学者，对 Visual Basic 代码的编写感到非常吃力，不用担心，后面我们将详细讲解编码的相关知识。Visual Basic 为用户编写代码提供了一个代码窗口，我们先讲 Code（代码）窗口。

1. Code 窗口

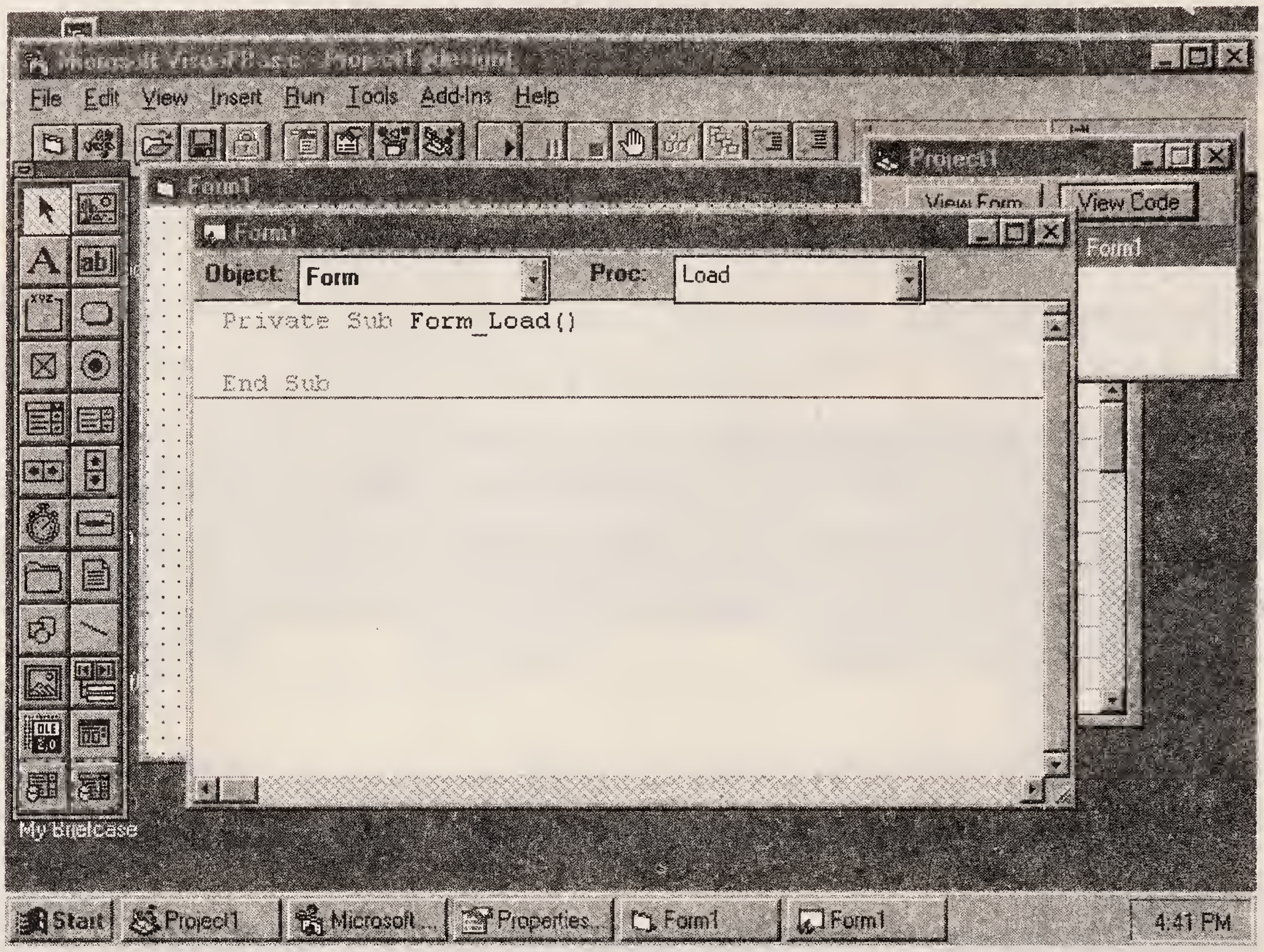
无论何时，用户双击控件或窗体均能打开代码窗口，用户也可以通过 View 菜单、F7 键或工程文件（Project）窗口中 View Code 按钮打开代码窗口，见图 5.13。打开 Code 窗口显示如图 5.14。


Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
Code F7
Form Shift+F7
Procedure Definition Shift+F2
Last Position Ctrl+Shift+F2
Object Browser... F2
Debug Window Ctrl+G
Project Ctrl+R
Properties F4
Toolbox
✓ Toolbar
Color Palette
1035,1170
6840×4635
View Form
View Code
Form1 Form1
Properties - Form1
Form1 Form
My Briefcase
Start
Project1
Microsoft ...
Properties - F...
Form1
4:14 PM

Auto3D	False
AutoRedraw	False
BackColor	&H80000005&
BorderStyle	2 - Sizable
Caption	Form1
ClipControls	True
ControlBox	True
DrawMode	13 - Copy Pen
DrawStyle	0 - Solid
DrawWidth	1
Enabled	True

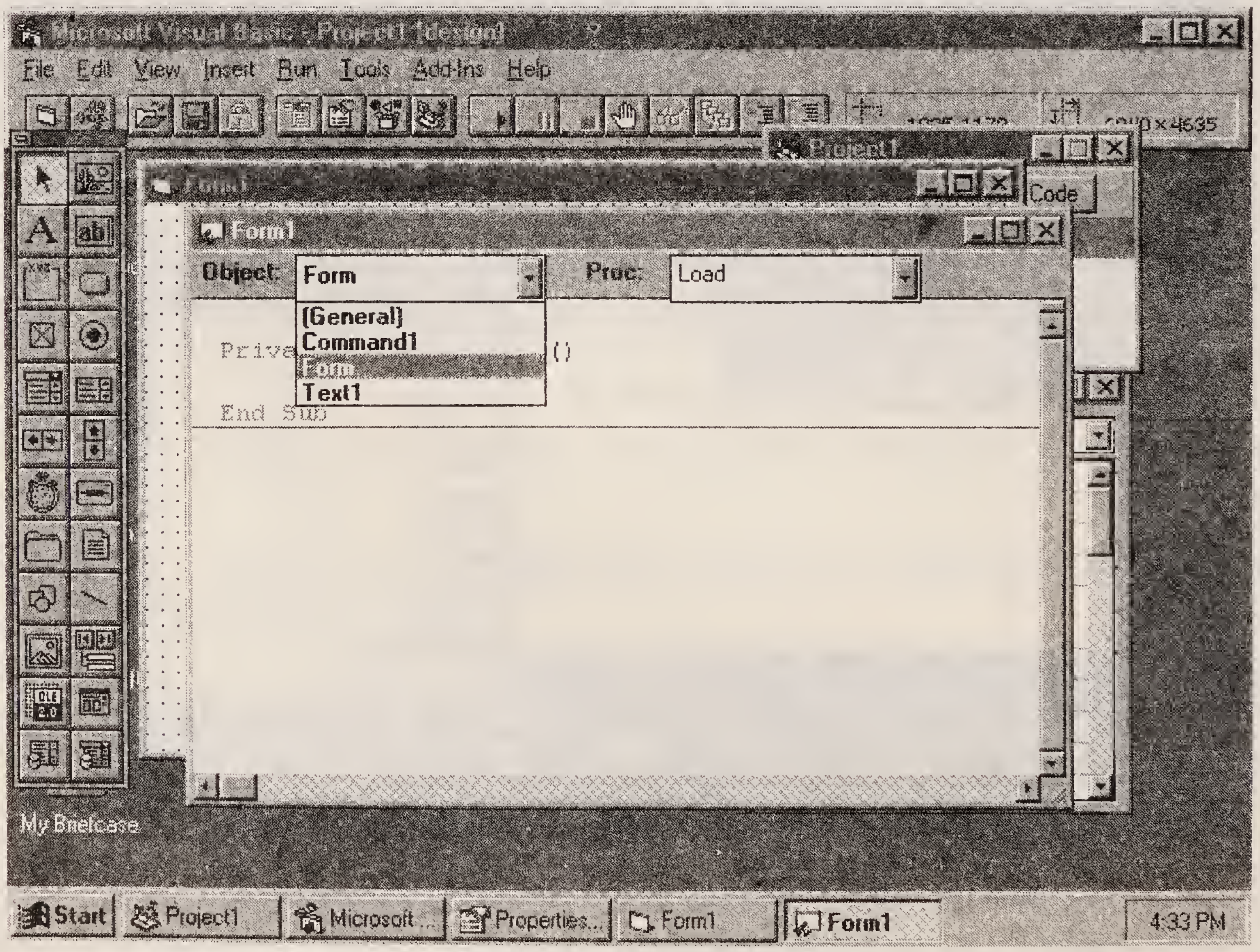
图 5.13　用 View 菜单或“View Code”按钮打开 Code 窗口

用户可以注意到 Code 窗口顶部有标有 Object 和 Proc 两个文字框。单击 Object 列表框旁箭头，该框列出当前用户窗体中所有对象，如果用户当前窗体中有文本框和命令按钮，则显示如图 5.15。其中有一个 general 的对象在后面章节讨论，此处用户不必关心。如果用户单击 Proc 列表框中向下箭头，会看见与 Object 框中被选对象的相关事件，见图 5.16，如果用户想选择对象或事件，单击它即可。


Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
Project1
View Form
View Code
Form1
Form1
Form1
Object:
Form
Proc:
Load
My Briefcase
Start
Project1
Microsoft ...
Properties...
Form1
Form1
4:41 PM

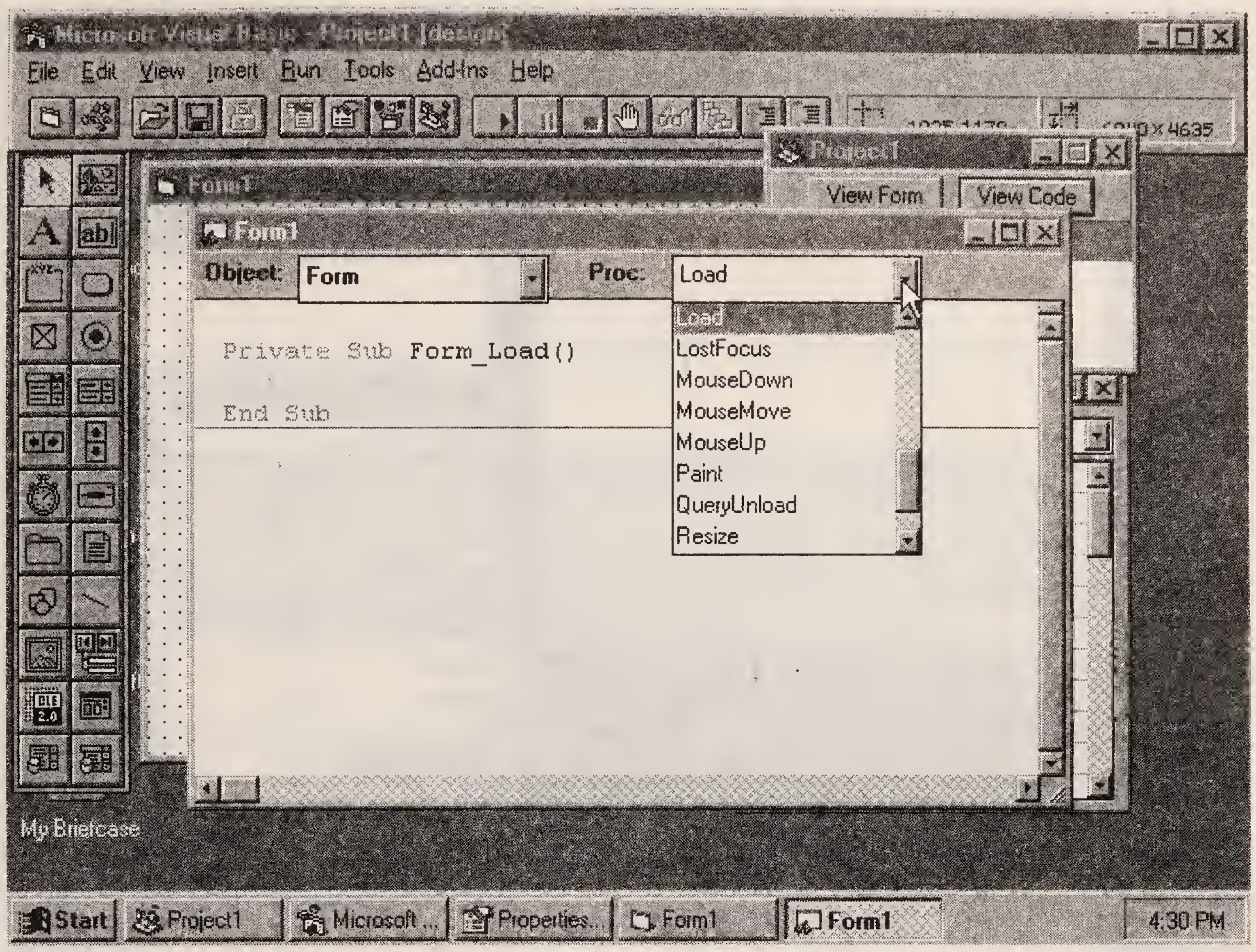
```
Private Sub Form_Load()

End Sub
```

图 5.14　Code 窗口


Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
Project1
Code
Form1
Form1
Object:
Form
Proc:
Load
(General)
Command1
Form
Text1
My Briefcase
Start
Project1
Microsoft ...
Properties...
Form1
Form1
4:33 PM

```
Priva                        ()

End Sub
```

图 5.15　当前所有对象



```
Private Sub Form_Load()

End Sub
```

Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
Project1
View Form
View Code
Form1
Form1
Object: Form
Proc: Load
Load
LostFocus
MouseDown
MouseMove
MouseUp
Paint
QueryUnload
Resize
My Briefcase
Start
Project1
Microsoft ...
Properties...
Form1
Form1
4:30 PM

图 5.16　被选对象的相关事件

在图 5.14 的 Code 窗口中还可以看到如下代码：

```
private Sub Form _ Load()
End Sub
```

该代码是 Visual Basic 的缺省调用事件过程，用于通过代码设置窗体或变量的初始化属性，有关这方面知识将在后面有关章节讲到，现在读者不必过份关心。

2. 窗体对事件的响应

现在编写一个窗体对单击事件响应并在文本框中显示一段信息的例子，用户要想完成此项工作可按下列步骤进行：

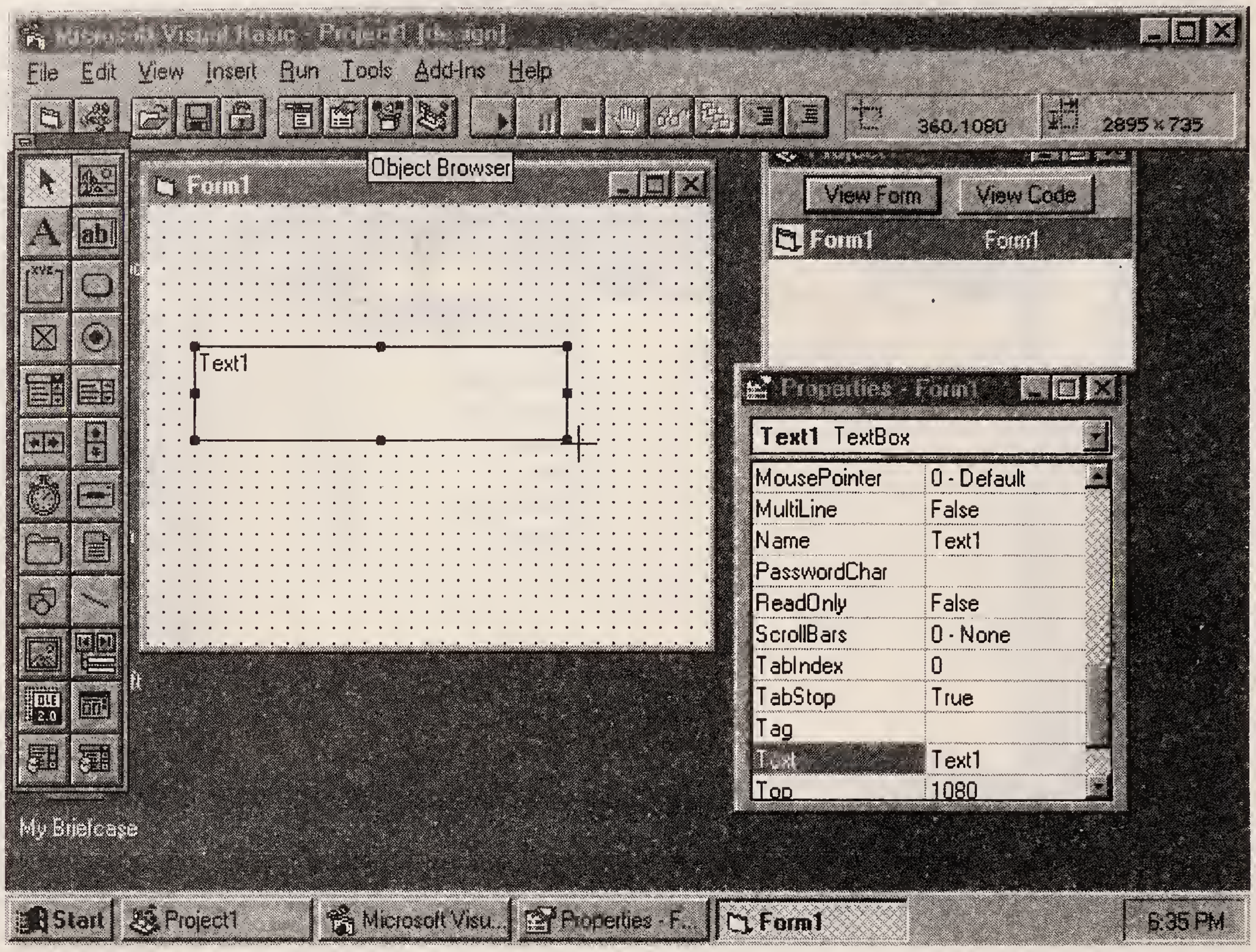
(1) 单击 File 菜单的 New Project 命令。

(2) 移动和改变窗体大小使工程文件窗口和属性窗口显示出来，并对窗体实现初步定制。

(3) 单击工具箱中的文本框图标，此时鼠标指针变“＋”标号，用左按钮拖动“＋”标号，然后释放鼠标即可显示文本框见图 5.17(后面一节将讲有关控件的具体操作)。

(4) 单击 Properties 窗口的标题条激活该窗口。

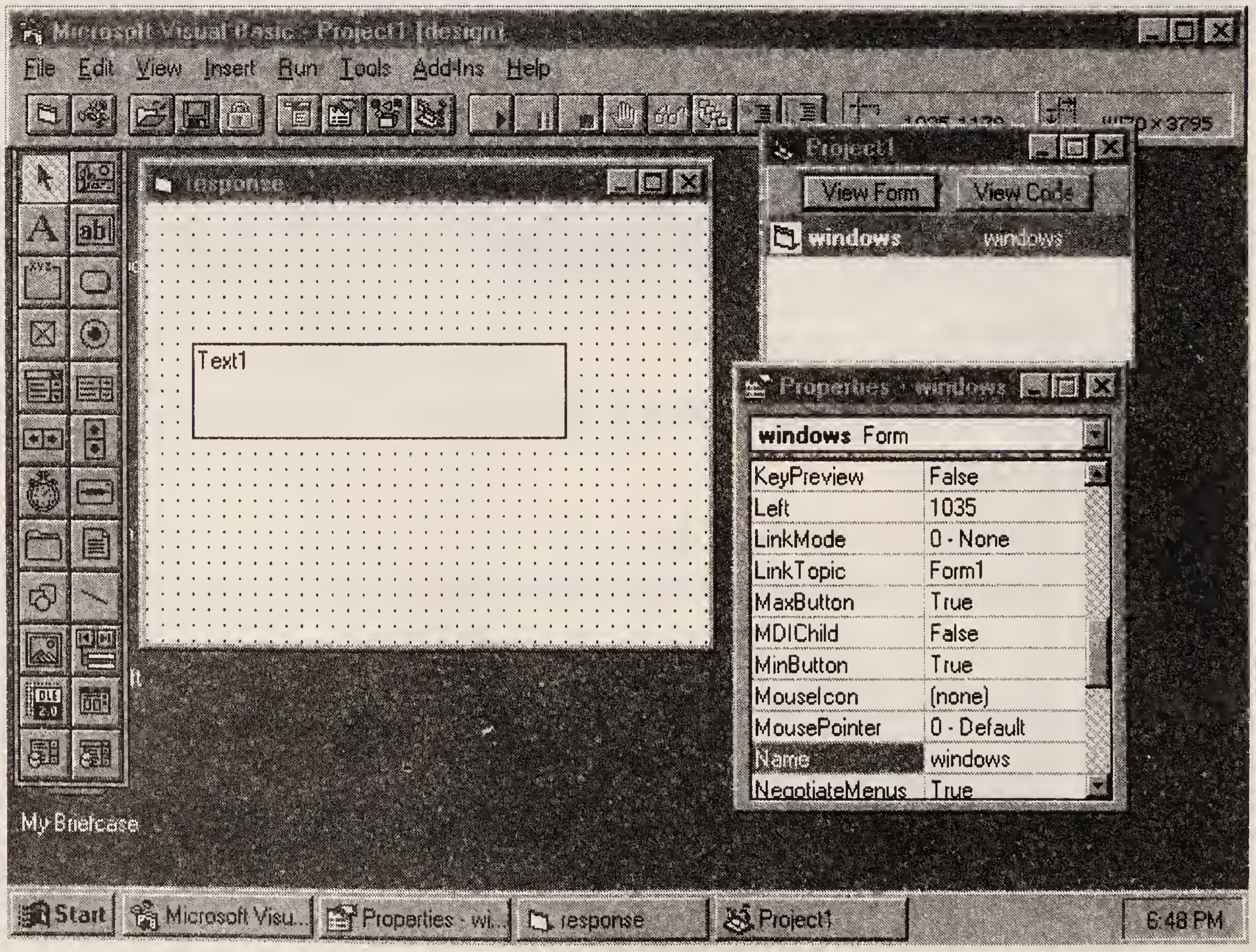
(5) 打开 Properties 窗口的对象列表框，选择 Forml 对象，然后分别单击属性列表中的 Name 属性和 Caption 属性，在设置框分别键入 Windows 和 response，前者代表在代码中窗体的名字，后者显示在窗体的标题条中，见图 5.18。用户可以看到窗体标题条中显示了 response，同时工程文件窗口中的 Form1 也变成了 Windows。


Microsoft Visual Basic - Project1 [design]
File
Edit
View
Insert
Run
Tools
Add-Ins
Help
360,1080
2895 x 735
Object Browser
Form1
Text1
View Form
View Code
Form1　Form1
Properties - Form1
My Briefcase
Start
Project1
Microsoft Visu...
Properties - F...
Form1
6:35 PM

Text1 TextBox

MousePointer	0 - Default
MultiLine	False
Name	Text1
PasswordChar	
ReadOnly	False
ScrollBars	0 - None
TabIndex	0
TabStop	True
Tag	
Text	Text1
Top	1080

图 5.17　建立文本框


Microsoft Visual Basic - Project1 [design]
File
Edit
View
Insert
Run
Tools
Add-Ins
Help
response
Text1
Project1
View Form
View Code
windows　windows
Properties - windows
My Briefcase
Start
Microsoft Visu...
Properties - wi...
response
Project1
6:48 PM

windows Form

KeyPreview	False
Left	1035
LinkMode	0 - None
LinkTopic	Form1
MaxButton	True
MDIChild	False
MinButton	True
MouseIcon	(none)
MousePointer	0 - Default
Name	windows
NegotiateMenus	True

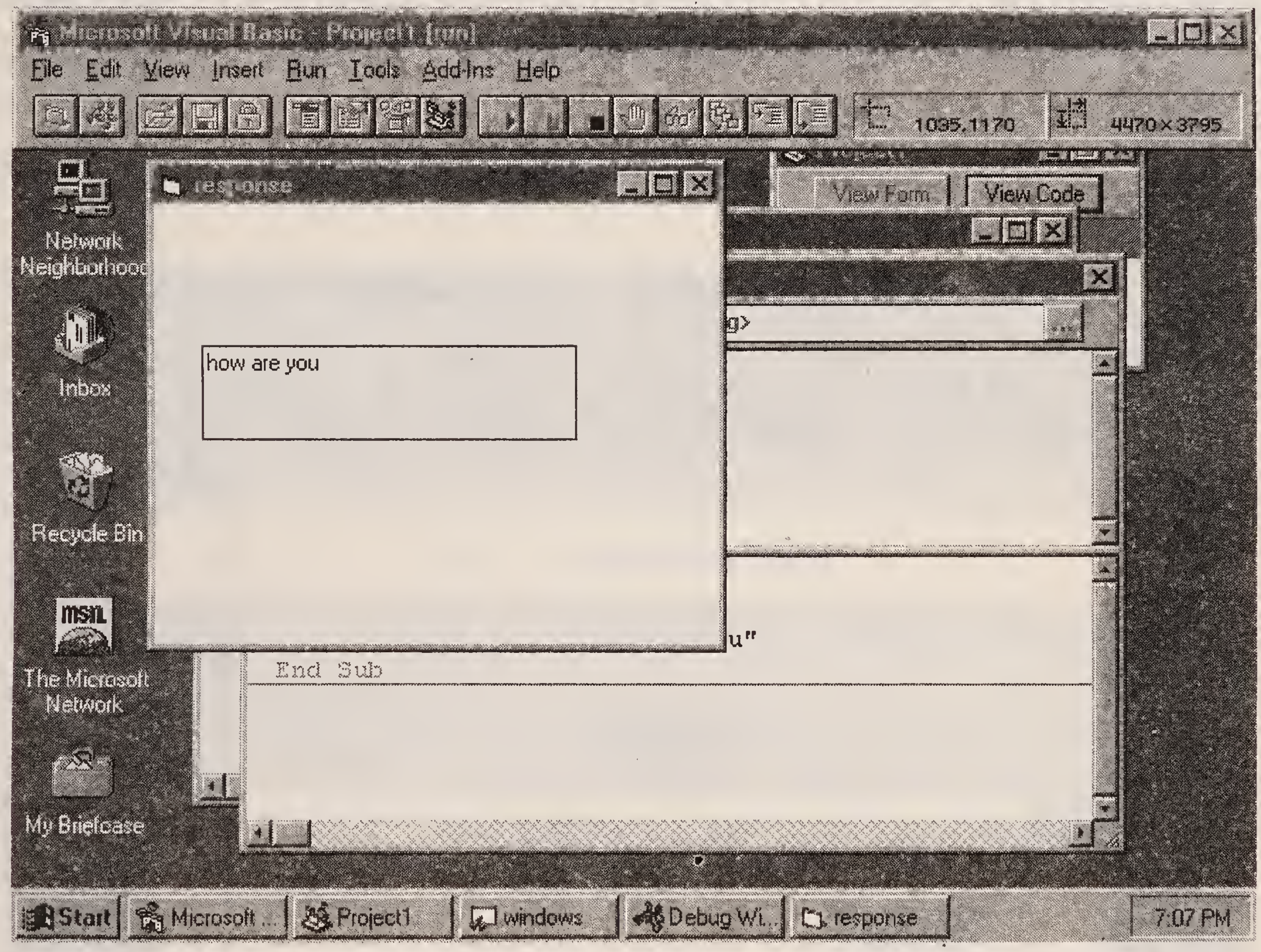
图 5.18　设置窗体属性

（6）同设置窗体属性一样，设置文本框的 Name 属性为 output text。

（7）双击窗体空白处打开 Code 窗口，选择窗体和相应的单击事件，编写以下代码：

```
Private Sub Form _ Click()
  outputtext. Text=" "
  outputtext . Text="How are you"
End Sub
```

（8）键入 F5 运行程序，出现一个 response 窗体，然后单击窗体空白处，用户可以看到文本框显示“How are you”，见图 5.19。



```
how are you
```

Microsoft Visual Basic - Project1 [run]
File Edit View Insert Run Tools Add-Ins Help
1035,1170
4470×3795
response
View Form
View Code
Network Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
End Sub
Start
Microsoft ...
Project1
windows
Debug Wi...
response
7:07 PM

图 5.19　程序运行结果

（9）单击 Run 菜单中的 End 命令或者单击工具条上的 End 图标就可以结束程序运行。

从这里用户可以看到用 Visual Basic 编写应用程序非常简单，用户在阅读以后有关章节以后会编写更为精彩的应用程序。

5.3　控件操作

前面用户已涉及到了一点有关控件操作知识，例如上节单击显示信息实例中文本框的生成，在那里用户或许感到有点迷惑，这节将详细介绍有关控件的操作。

5.3.1　控件的生成

今后用户在编写应用程序的时候经常会遇到把有关控件加到窗体，把控件加到窗体的方法有两种：

(1) 用鼠标单击工具箱中所需要的工具，工具图标的颜色变暗标志工具已被选中，此时鼠标指针在窗体中变成小十字号，把鼠标指针放到窗体中用户想放置控件的地方，按下鼠标左按钮拖动鼠标就能在窗体中放置控件。

(2) 用鼠标左按钮双击工具箱中用户所需的控件，这样就可以把控件放置到窗体的中央。

提示　在控件被放置到窗体上后，控件周围出现八个小黑体方块(称调节杆)，这表明该控制处于激活状态，单击窗体空白处小黑体方块消失，控件处于非激活状态。

控件被放置到窗体上以后，用户可能因某种需要而要求重新放置控件的位置或者改变控制的大小。如果用户确有这种需要可单击该控件使之处于激活状态，然后按住鼠标左按钮拖动控件到用户满意的位置上放置就可实现控件位置的重新安排。用户把鼠标指针指向调节杆，此时指针变为带双箭头的标记，然后按住鼠标左按钮拖动鼠标把控件大小改变到满足用户需要为止，见图 5.20。

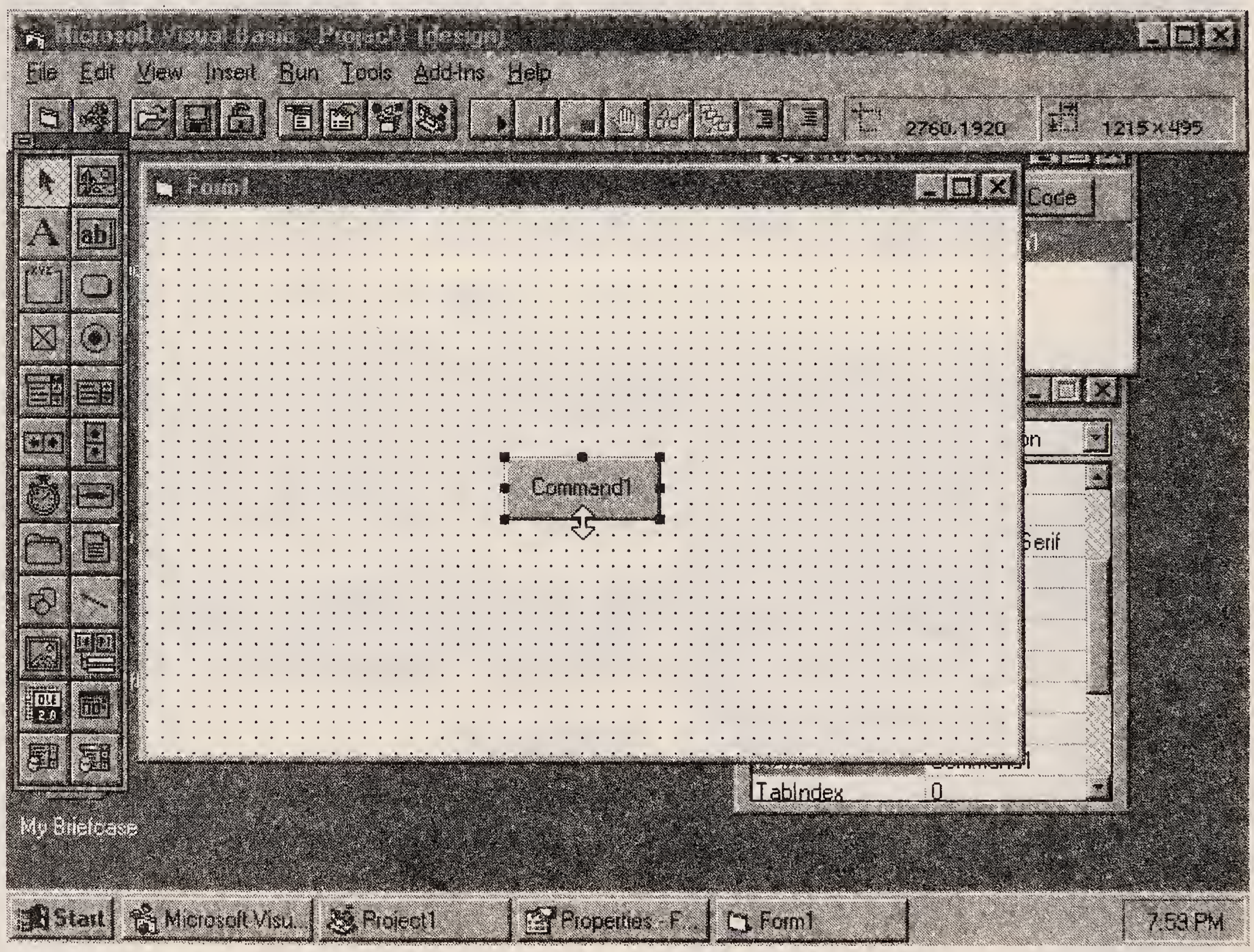

Microsoft Visual Basic - Project1 [design]
File
Edit
View
Insert
Run
Tools
Add-Ins
Help
2760,1920
1215 x 495
Form1
Code
Command1
Serif
TabIndex
0
My Briefcase
Start
Microsoft Visu...
Project1
Properties - F...
Form1
7:53 PM

图 5.20　改变控件大小

5.3.2　同时操作多个控件

为了提高工作效率，有时用户需要同时移动多个控件，这时用户只需用鼠标左按钮单击控件的同时按住 Ctrl 键逐个选择即可选中一组控件，这些控制都处于激活状态，见图 5.21。然后用户移动其中一个控件对其他控制也造成相同影响，见图 5.22。

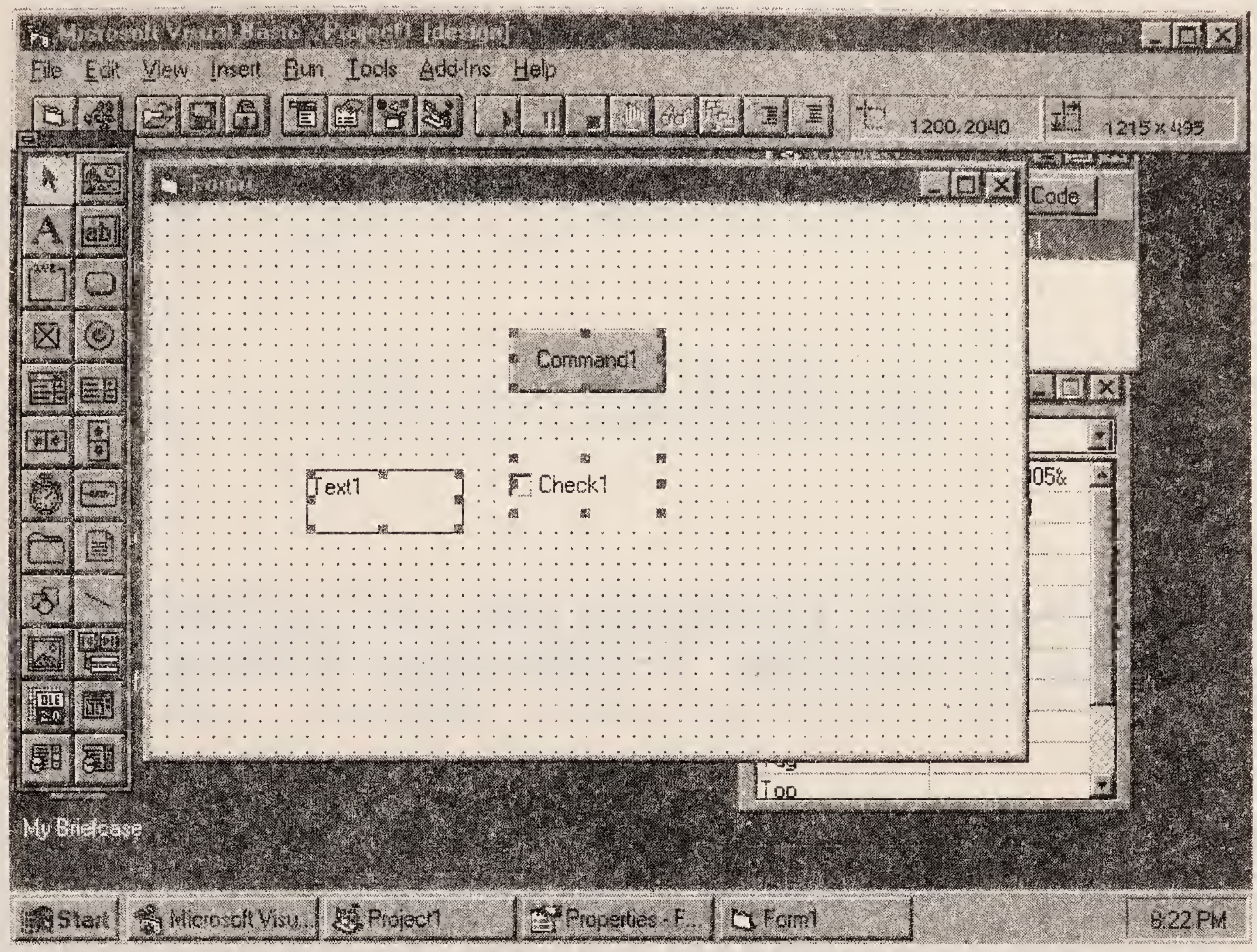

Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
1200, 2040
1215 x 495
Form1
Code
Command1
Text1
Check1
Top
My Briefcase
Start
Microsoft Visu...
Project1
Properties - F...
Form1
8:22 PM

图 5.21　同时激活多个控件

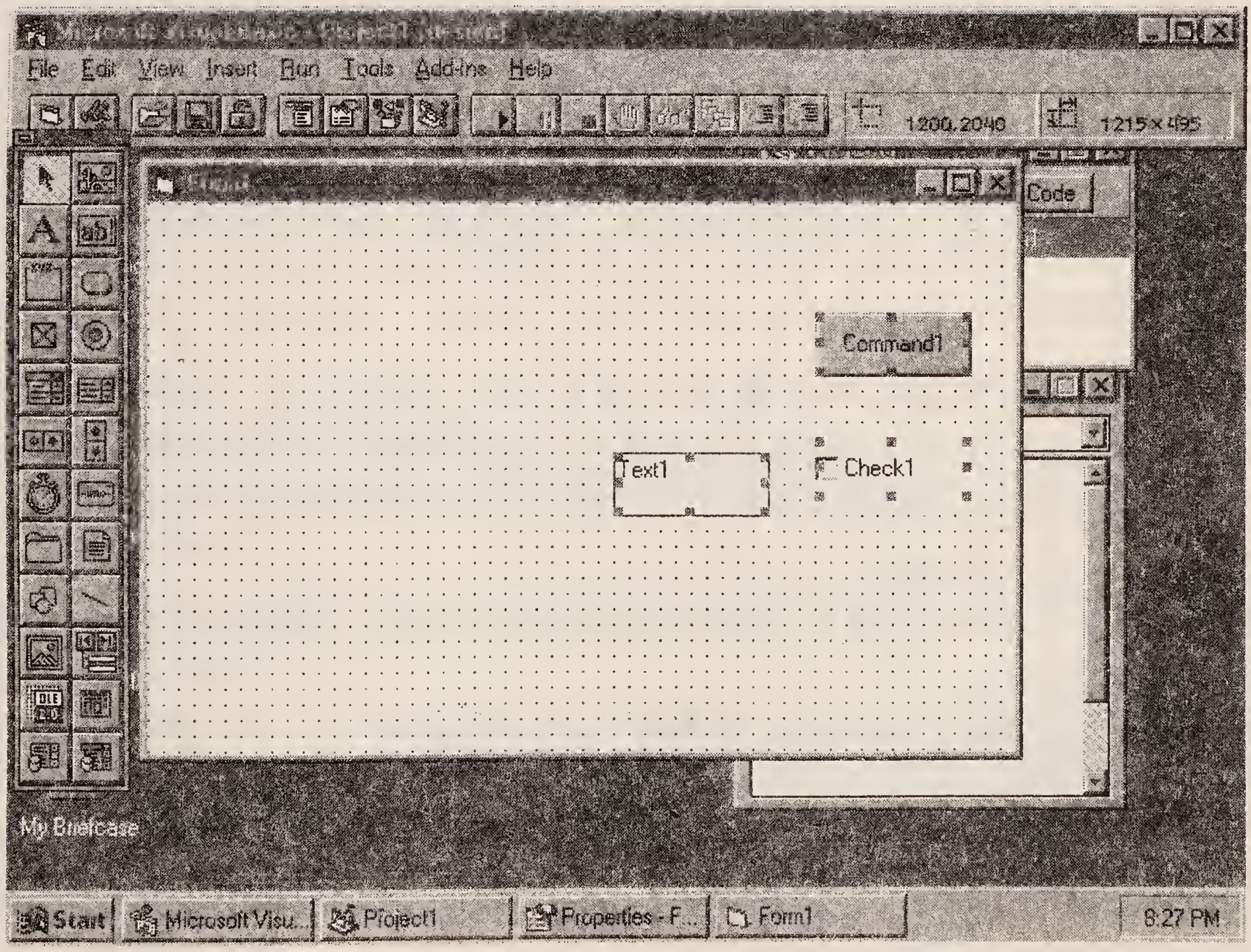

Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
1200, 2040
1215 x 495
Form1
Code
Command1
Text1
Check1
My Briefcase
Start
Microsoft Visu...
Project1
Properties - F...
Form1
8:27 PM

图 5.22　同时移动多个控件

5.3.3　控件的删除

用户如果想删除窗体上有关控件必须先激活该控件，然后单击 Edit 菜单的 Clear 命令或者在键盘上键入 Del 键。如果用户想同时删除多个控制必须按前述方法同时选取多个控件。

5.4　控件的属性及相关事件

5.4.1　标签

标签控件是在需要显示用户无法直接修改的文本时出现，当用户想标记窗体中某控件或显示用户程序生成的文本时使用标签。打开标签的属性窗口用户可发现以下属性：

1. Autosize

该属性有两种设置值：True 和 False。如果用户把该属性设置为 True，则标签控件大小随文本的改变而变化。如果该属性被设置为 False（缺省设置），则标签控件不因文本的改变而变化。当程序运行时想改变标签内容则应设置为 True，用户应该记住在窗体中为标签留出位置。

2. BorderStyle

在窗体中已涉及到该属性，不过对标签控件来说它只有两种设置值：0 和 1，其中 0 是缺省值。如果该属性被设置为 0 则标签没有边框。如设置成 1，标号将有单线边框。

3. WordWrap

该属性也有两种设置情况：True 和 False，其中 False 是缺省设置。如果设置为 True，则标签控件将在垂直方向变化大小以和文本相适应，但将限制其原先的水平大小。如果该属性被设置为 False，标号不会改变垂直方向大小以适应文本的需要，但水平方向上的大小改变则取决于 Autosize 的设置情况。

标签还有诸如字体等其他属性，这和前面讲到的窗体相似，因篇幅限制，在此不多讲了，用户可以参考有关手册。

同标签最常见的相关事件是单击、双击和 Change 等事件。

5.4.2　文本框

文本框一般用于从用户获取输出或显示信息，文本框有时又叫做“编辑区”或“编辑控件”。文本框有以下常用的属性。

1. MaxLength

该属性决定文本框中输入的数目是否有长度限制，缺省值 0 表明对单行文本框，只受操作者系统内存的限制，若该属性被设置为大于 0 的数，该数表示能够输入的最大字符数。

2. Multiline

该属性表明文本框是否可以接收和显示多行文本，仅在运行时可读。该属性有两种设置值：True 和 False。True 设置允许有多行文本，False（缺省）设置忽略单行时的回车换行和限制数据，即仅在一行显示或接收一小片文本。

3. PasswordChar

该属性提示用户在文本框中输入口令，其缺省值为空字符串。

4. ScrollBars

该属性指定文本框是否有水平或垂直滚动条，它有四种设置值：0、1、2、3。0 设置（缺省）表示没有水平或垂直滚动条，1 表示有水平滚动条，2 表示有垂直滚动条，3 表示既有水平滚动条又有垂直滚动条。对于设置值为 1，2 或 3 的文本框，用户必须把 Multiline 属性设置为 True。

5. SelLength、SelStart、SelText

SelLength 属性确定选定的字符数。Selstart 属性确定选定文本的起始点，如果没有选定文本，则确定插入点的位置。SelText 属性指定当前选定文本的字符串，如果没有选定字符，则包含一个空串。对于 SelLength 和 SelText，设置值的有效范围为 0 到文本长度，即文本框中编辑区的字符总数。对于一些任务，可以使用这些特性。例如，设置插入点，建立插入范围，在文本框中选定原子串或清除文本等，与剪贴板一起使用有助于拷贝、剪切及粘贴等操作。当用户使用这些特性时，Sellength 的设置小于 0，会引起运行时的错误。SelStart 设置值比文本长度大时，可向已存在的文本设置特性，SelStart 的变化会改变插入位置并将 SelLength 设置为零。SelText 设置值为一个新值时，设置 SelLength 为 0 并用新字符串代替选定的文本。

注意　该属性也适用于组合框。

与文本框相关的事件除 Click（单击）、DblClick（双击）之外常见还有以下几种：

(1) Change 事件

此事件是改变文本框的内容，当用户改变文本或用代码改变 Text 属性时发生此事件。

(2) GotFocus 事件

当文本框收到焦点，无论用户如何动作，或敲击、或单击对象或使用 SetFocus 改变代码中的焦点时，该事件都会发生。用户必须注意：一个文本框只有当它被激活和可见性为 True 时，才能接收到焦点。该事件还和窗体、复选框、组合框、命令按钮、目录列表框、驱动列表框和文件列表框等对象相关。

(3) LostFocus 事件

当文本框失去焦点时，用户可以记下这个对象或单击另一个对象，也可以用 SetFocus 改变焦点代码。该事件同 GotFocus 事件一样也同窗体、复选框和组合框等对象相关。

5.4.3　命令按钮

用户想开始、中断或结束一个过程时要选择命令按钮，被选择时，命令按钮显示按下状态，所以有时被称作下按“按钮”。以下是命令按钮最重要的属性。

1. Cancel

该属性决定窗体中的命令按钮是否是Cancel按钮。如果该属性被设置为True，则命令按钮是取消按钮，如设置为False，则命令按钮不是取消按钮。窗体中唯一的命令按钮可以是Cancel按钮。当一个命令按钮的Cancel属性被设置为True时，窗体中其他命令按钮的Cancel属性会自动被设置为False，并且当此窗体是活动窗体时，用户可通过单击它，按Esc键或当此按钮有焦点时按Enter键来选中此命令按钮。

2. Caption

该属性设置显示在命令按钮表面上的文本，用户可以在想用作热键的字母前加一&符号。例如用&Start作为标题将给出一个按钮文本为Start，按Alt+S键选中Start按钮。

3. Default

该属性把窗体中哪个命令按钮作为缺省按钮。如果该属性被设为False，则此命令按钮不是缺省按钮。如果该属性被设置为True时，则此命令按钮为缺省按钮。窗体中只允许有一个按钮是缺省按钮。当一个按钮的Default设置为True时，窗体中其他按钮的Default被自动设置为False。在很多情况下把OK按钮的Default属性设为True，这时用户可键Enter键来选择OK按钮。

命令按钮的常见事件是Click、GotFocus和LostFocus。

5.4.4　框架

用户可利用框架控件把其他一些相关控件组织在一个框内，要组成控件，首先绘制框架，然后把控件绘制在框架中，这样可以一起移动框架和控件。框架最常见的属性有以下几种。

(1) Caption

该属性给框架设定一个标题以使用户明白框架内的控件的用途。

(2) Enabled

该属性有两种设置：True和False。True允许用户存取框架中的控件，如果不想让用户存取组中的控件则应设置为False。

同框架最常用的相关事件是单击和双击。

5.4.5　单选按钮

单选钮是一个能开能关的控件，通常单选钮用于组成选项组，而这些选择项中只能选择一个。可通过一个框架、一个图片框或直接在一个窗体上画出它们来组合单选钮。要在一个框架或一个图片框中组合单选钮，先画出框架或图片框，然后在里面画单选按钮。单选按钮

常见属性有几下几种：

（1）Value 属性

该属性确定单选钮的状态。如果设置为 True，则单选钮被选中。如设置为 False，则单选钮没有被选中。

（2）Caption 属性

该属性显示单选钮附近的文本。

（3）Enabled 属性

用户可通过设置 Enabled 属性为 False 来禁止一个单选按钮。

同单选按钮相关的事件有 Click、GotFocus 和 LostFocus。

5.4.6　复选框

复选框在被选中时出现“X”号，当再被单击时消除复选框中的“X”号。使用此控件可以让用户选择 True/False 或 Yes/No。可在一组中放置多个复选框来显示多种选择，用户可以从中选择一个或多个，然而单选按钮每次只能选择一个，这是两者的不同之处。复选框常用属性和单选钮差不多，主要是 Caption、Value 和 Enabled，在此不再介绍，用法可参考单选钮中的用法。

同复选框最常用的相关事件是 Click、GotFocus 和 LostFocus。

5.4.7　列表框

用户可以从列表框中的一系列选项中选择一个或多个选项，如果选项的数量超过可显示的，列表框会自动增加滚动条，空列表框是没有什么作用的，因而用户必须填充列表框。列表框可以是单列或多列的，以下是用户必须知道的属性。

1. Columns

该属性决定列表框是水平滚动还是垂直滚动。如果是水平滚动，该属性决定显示几列。如果该属性被设置 0（缺省），则数据项安排一列，列表垂直滚动。如果该属性被设置 1 到 n 的整数，数据项被安排成串形，先填充第一列，然后是第二列，以此类推，列表框为水平滚动并且显示指定数量的列。

2. MultiSelect

该属性决定用户是否能够设置多个选择项以及多项选择如何实现。如果该属性被设置为 0（缺省），则不允许有多项选项。该属性被设置为 1，则允许有简单多项选项。该属性被设置为 2 时，则扩展多项选择，用户按 Shift＋单击或 Shift＋箭头键可把先前选项扩展到当前选项。

3. Sorted

该属性指定列表框的元素按字母顺序自动排列。该属性有两种设置值：True 和 False。如果被设置为 True，则列表项按字母顺序排列。如果被设置为 False（缺省），则列表项不按字母顺序排列。

同列表框常见的相关事件有 Click、GotFocus 和 LostFocus。

5.4.8　组合框

组合框综合了文本框和列表框的功能。使用此控件，可让用户选择在文本框中部分键入文字或从列表框部分选取项目。

1. Sorted

同列表框一样，该属性决定是否按字母顺序排列组合框中的项目。

2. Style

该属性指定组合框的类型和列表框部分的状态。如果该属性被设置为 0（缺省），则组合框类型为下拉组合框，包括一个下拉列表和一个编辑区，用户可以从列表中选择或在编辑区中显示信息。如果该属性被设置为 1，则组合框类型为简单组合框，包括一个编辑区和一个总显示列表，用户可以从列表选择或在编辑区键入信息。简单组合框的尺寸包括编辑和列表部分，默认的简单组合框尺寸最小，所以不显示任意列表，用户可增加 Height 属性来显示更多的列表。如果该属性被设置为 3，则只允许用户从下拉列表中做选择。

同组合框常用到的相关事件有单击，双击，Change、GotFocus LostFocus 等。

5.4.9　水平和垂直滚动条

滚动条提供一个在许多项目中或大量信息中简便移动的方法，它们也能提供当前位置的模拟表示，因此用户可使用滚动条作为输出设备、速度或数量显示器。该滚动条控件不同于出现在列表框文本中和其他控件中的滚动条。滚动条控件常见属性如下：

（1）Min，Max

Min 属性决定滚动块位置的最小值设置，即滚动块在最顶位置（垂直滚动条）或最左位置（水平滚动条）。Max 属性决定滚动块位置最大值设置，即滚动块在最底位置（垂直滚动条）或在最右位置（水平滚动条）。对每个属性可以指定－32768 到 32768 之间的一个整数，缺省设置是：Max —— 32767，Min —— 0。

（2）SmallChange，LargeChange

LargeChange 属性用于确定当用户单击滚动箭头和滚动框之间区域时，滚动条控件 Value 特性的改变数量。SmallChange 用于确定当用户单击滚动箭头时，滚动条控件中 Value 属性的改变数量。这两个属性都可以指定为 1 到 32767 之间的一个整数，如果缺省，则每个属性都设置为 1。

（3）Value

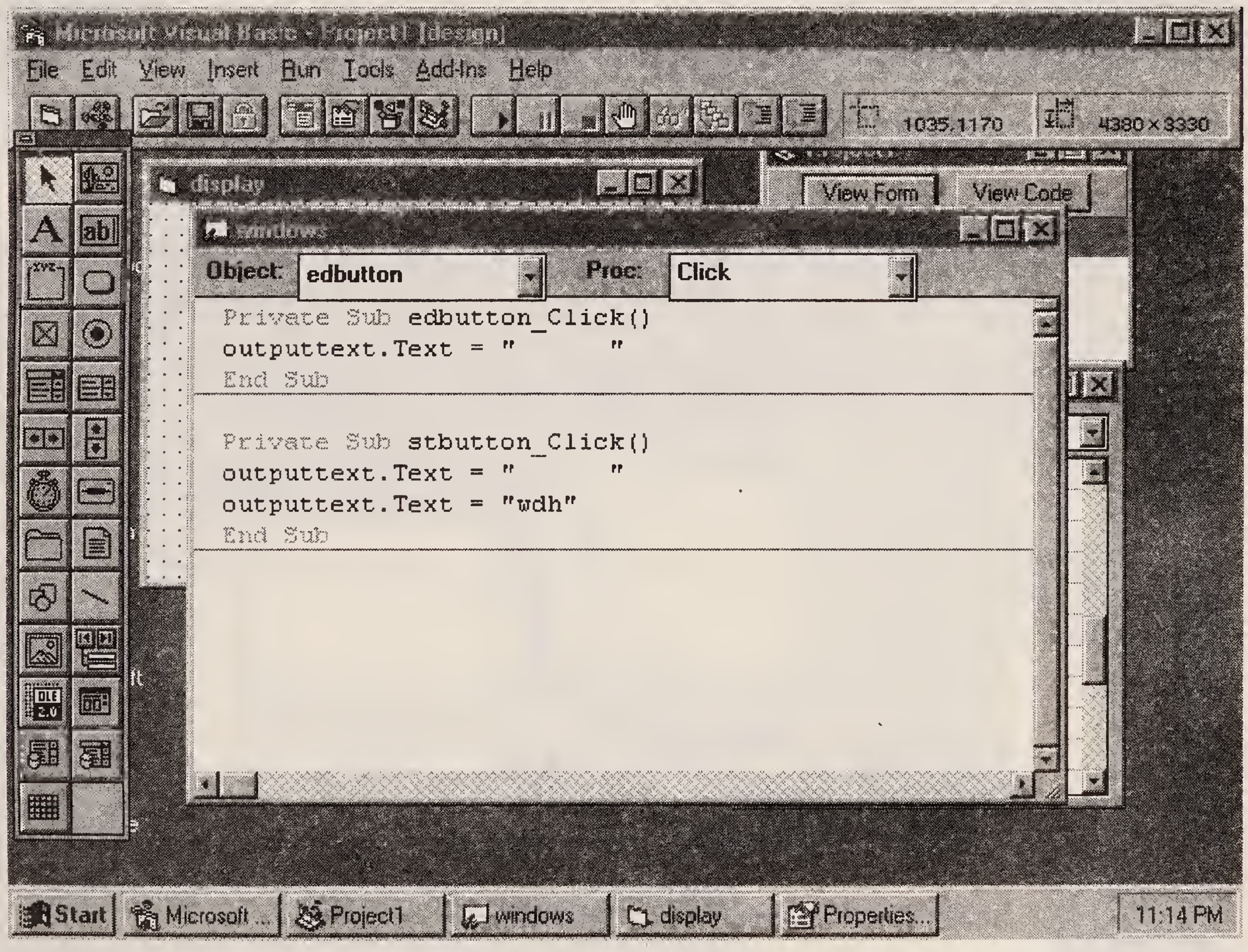
该属性确定滚动条当前位置，返回值总是介于 Max 属性和 Min 属性之间，它的设置值介于－32768 和 32767 之间。

同滚动条相关的事件除 Change、GotFocus 和 LostFocus 之外常见的还有 Scroll 事件，当用户在滚动条上拖动控件时发生。用户可利用这个事件去执行计算或管理正在滚动条中变化的控件，与此相对，在滚动条变化后，用户想发生一次更新时使用 Change 事件。

为了减轻读者的负担，本节不想把所有的控件都讲一遍，剩下的控件分别分散在后面有关章节讲解，例如画线、形状及图像控件将在绘图操作一章讲解，驱动器列表框、目录列表框和文件列表框将在文件操作一章讲解，公用对话框控件将在对话框设计一章讲解。OLE2.0 控件和 Data 控件将在第十二章讲解。

5.5　控件对事件的响应

本节将以命令按钮和文本框为例讲控件对事件响应的一个简单实例。当用户单击开始按钮时，文本框中显示信息。当用户单击结束按钮时文本框中信息消失。用户先用 Properties 窗口把窗体和两个按钮的 Name 属性分别设置为 Windows、Stbutton，edbutton，它们的 Caption 属性分别为 display、Start、end。然后用户打开 Code 窗口。在窗口编写代码，见图 5.23。

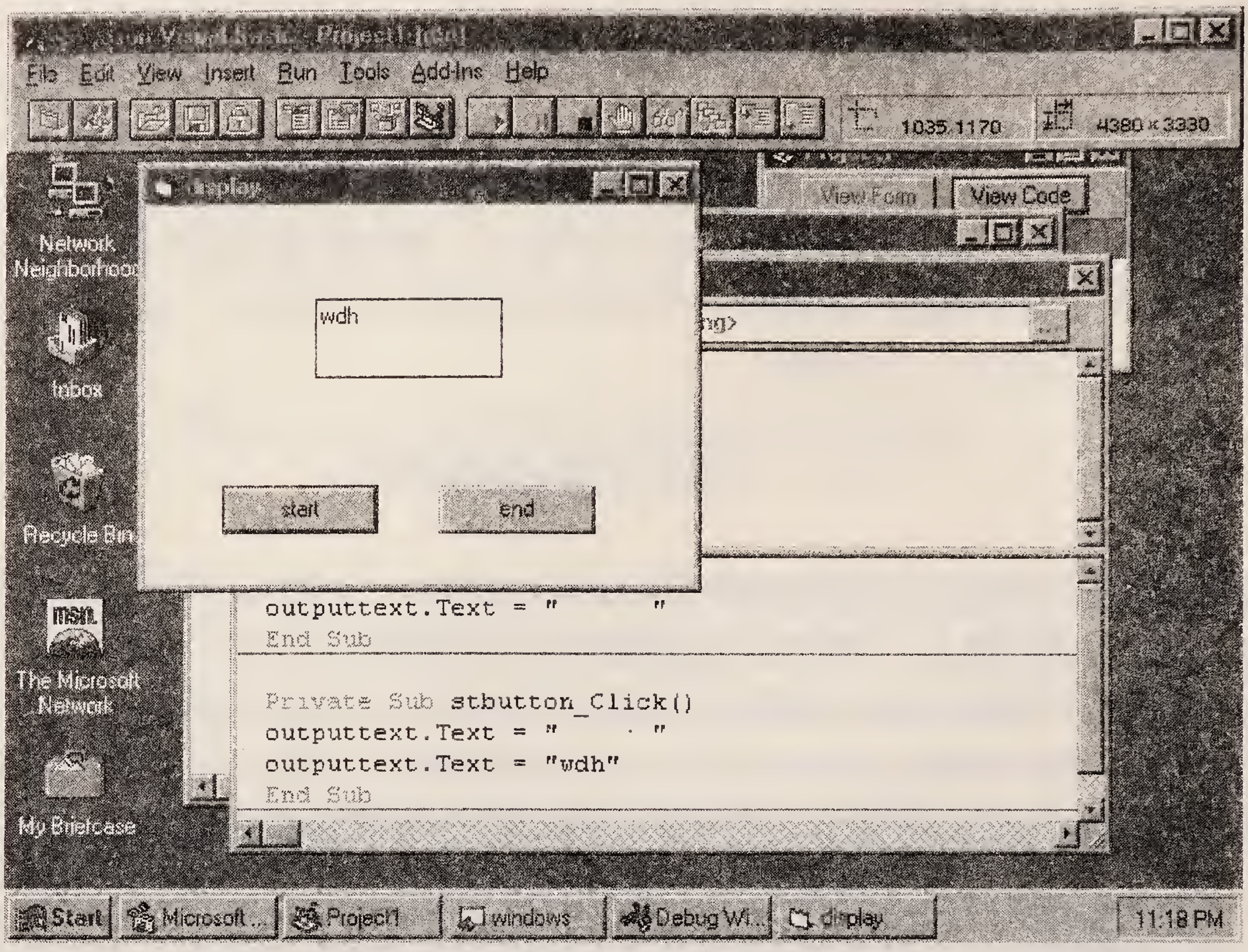

Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
1035,1170
4380 x 3330
display
View Form
View Code
windows
Start
Microsoft ...
Project1
windows
display
Properties...
11:14 PM

Object: edbutton　Proc: Click

```
Private Sub edbutton_Click()
outputtext.Text = "      "
End Sub

Private Sub stbutton_Click()
outputtext.Text = "      "
outputtext.Text = "wdh"
End Sub
```

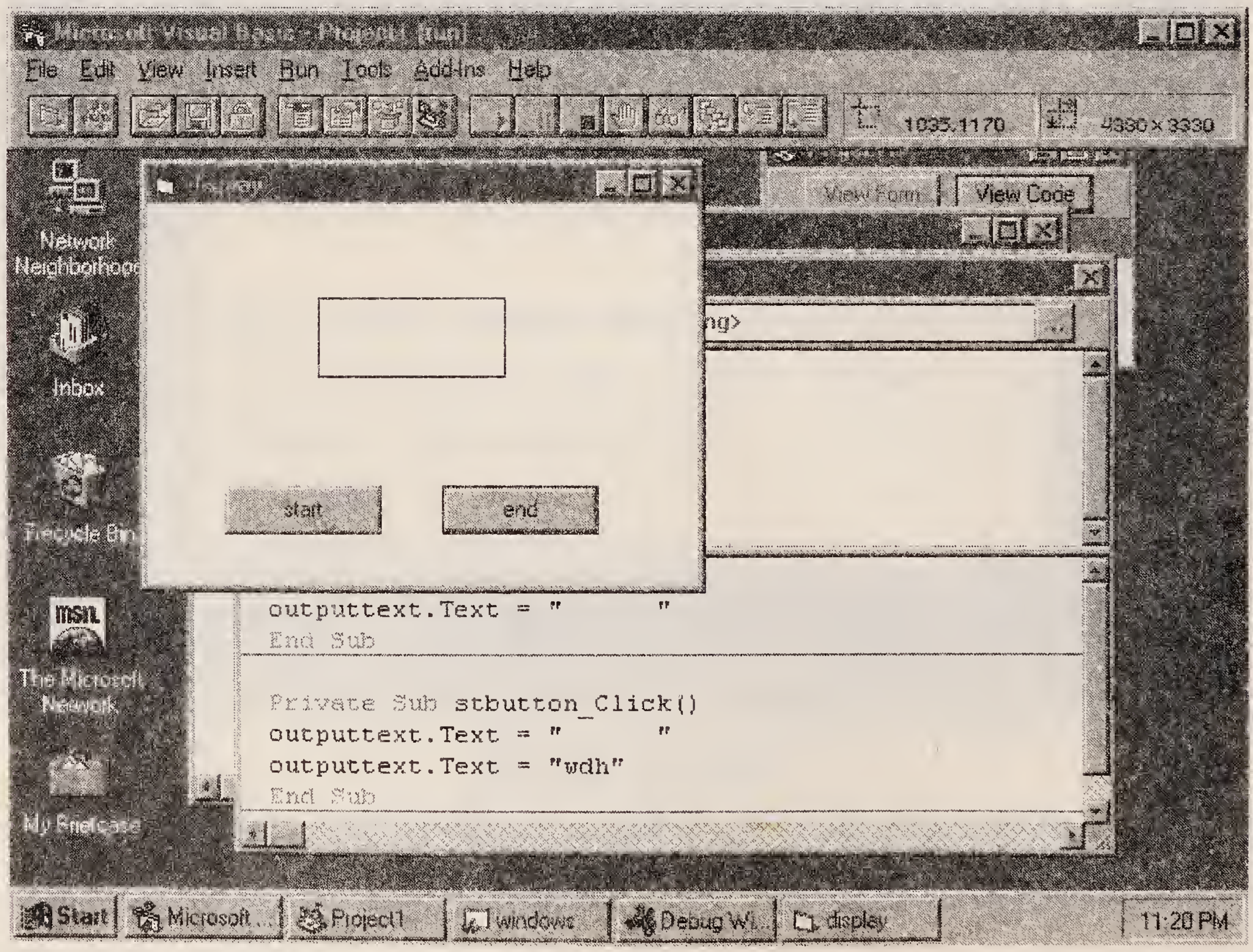
图 5.23　编写代码

用户键入 F5 键即可运行，单击 Start 按钮显示结果如图 5.24。单击 end 按钮显示结果如图 5.25，用户可用 Run 菜单的 End 命令结束程序的运行。


File Edit View Insert Run Tools Add-Ins Help
1035,1170
4380 x 3330
display
wdh
start
end
View Form
View Code
Network Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Start
Microsoft...
Project1
windows
Debug Wi...
display
11:18 PM

```
outputtext.Text = "        "
End Sub

Private Sub stbutton_Click()
outputtext.Text = "       "
outputtext.Text = "wdh"
End Sub
```

图 5.24　单击 Start 按钮


Microsoft Visual Basic - Project1 [run]
File Edit View Insert Run Tools Add-Ins Help
1035,1170
4380 x 3330
display
start
end
View Form
View Code
Network Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Start
Microsoft...
Project1
windows
Debug Wi...
display
11:20 PM

```
outputtext.Text = "        "
End Sub

Private Sub stbutton_Click()
outputtext.Text = "        "
outputtext.Text = "wdh"
End Sub
```

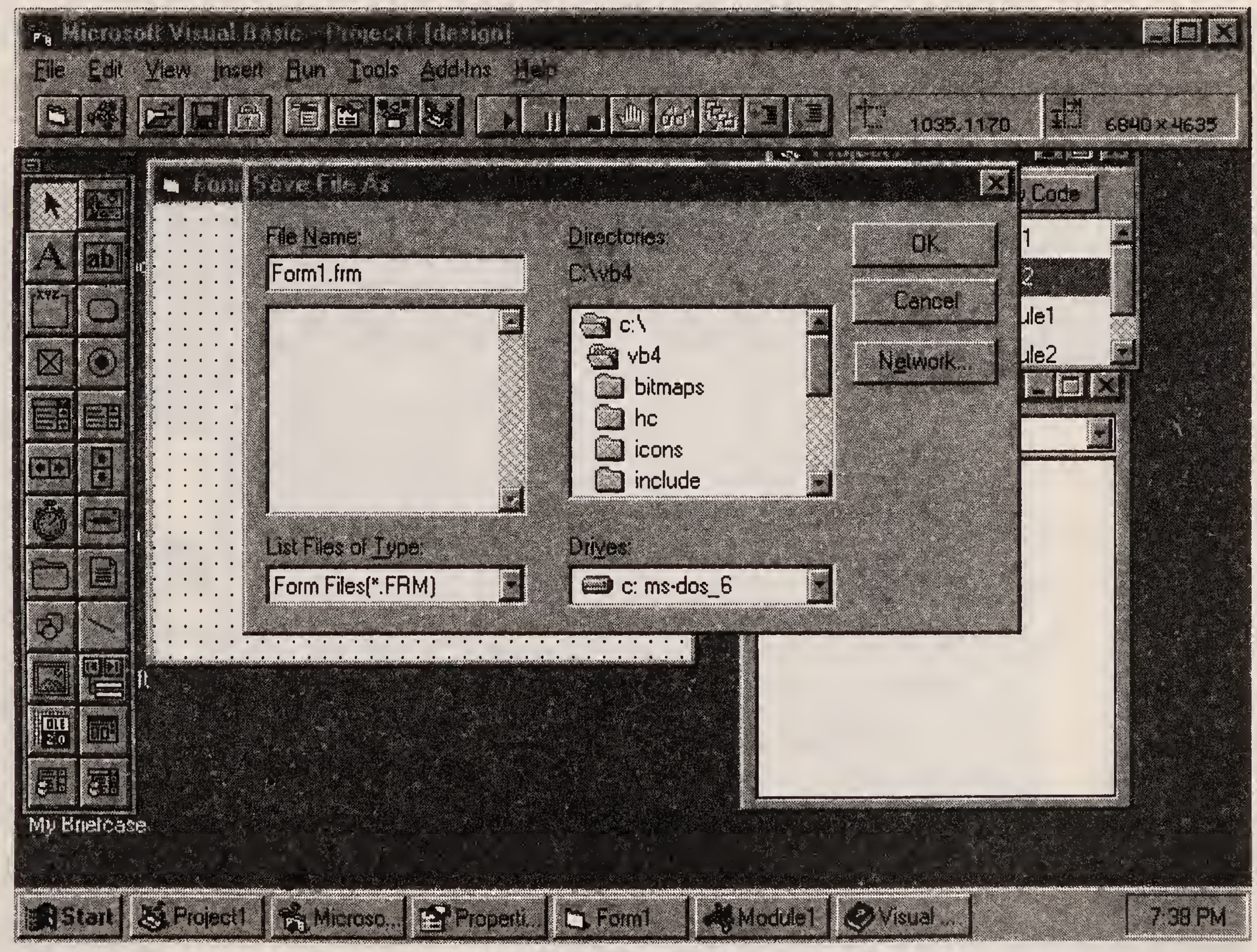
图 5.25　单击 end 按钮

5.6　存储用户的工作

用户应当养成随时存储自己所做的工作习惯，当用户退出一个应用程序时，Visual Basic 将询问用户是否存储用户所做的工作。Visual Basic 4.0 中提供了存储用户工作的多种方法。用户能够像存储一个特定的窗体一样也能够存储整个工程文件。

5.6.1　存储一个新的工程文件

用户从 File 菜单单击 Save Project 命令或者单击工具条上的存储快捷键。如果用户以前没有存储该工程文件，则 Visual Basic 将打开一个对话框，见图 5.26。在 File name 下面用户可看到窗体名和扩展名 Frm，用户可在驱动器列表和目录列表中选择存储文件的路径，按回车键 Visual Basic 将窗体上的所有信息及相关代码存储起来。如果用户用已存在的文件名存储文件，当用户选择 OK 时，Visual Basic 将询问用户是否覆盖已存在的文件。如果用户文件中除窗体之外还有标准模块或类模块，则 Visual Basic 将出现相似的对话框，并且以缺省扩展名 BAS 或 CLS 存储相应的模块。然后 Visual Basic 将出现询问. MAK 文件名，其缺省名为 Project1. MAK。如果用在开发和改进工程文件的过程中用了 Save Project 命令存储对同一个工程文件所作的修改，则 Visual Basic 不再向用户提示对话框。


Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
1035,1170
6840 x 4635
Save File As
OK
Cancel
Network...
c:\
vb4
bitmaps
hc
icons
include
Code
My Briefcase
Start
Project1
Microso...
Properti...
Form1
Module1
Visual ...
7:38 PM

File Name: Form1.frm

Directories: C:\vb4

List Files of Type: Form Files(*.FRM)

Drives: c: ms-dos_6

图 5.26　存储新工程文件时输入一个文件名

5.6.2　用新的文件名存储已存在的工程文件

如果用户想更改已存在工程文件的名字，则可以单击 File 菜单的 Save Project As... 命令，这时出现同图 5.26 一样的对话框，用户可以更改文件名。如果需要用户还可以通过驱动器列表和目录列表更改文件的存放路径。

5.6.3　存储新的窗体、标准或类模块

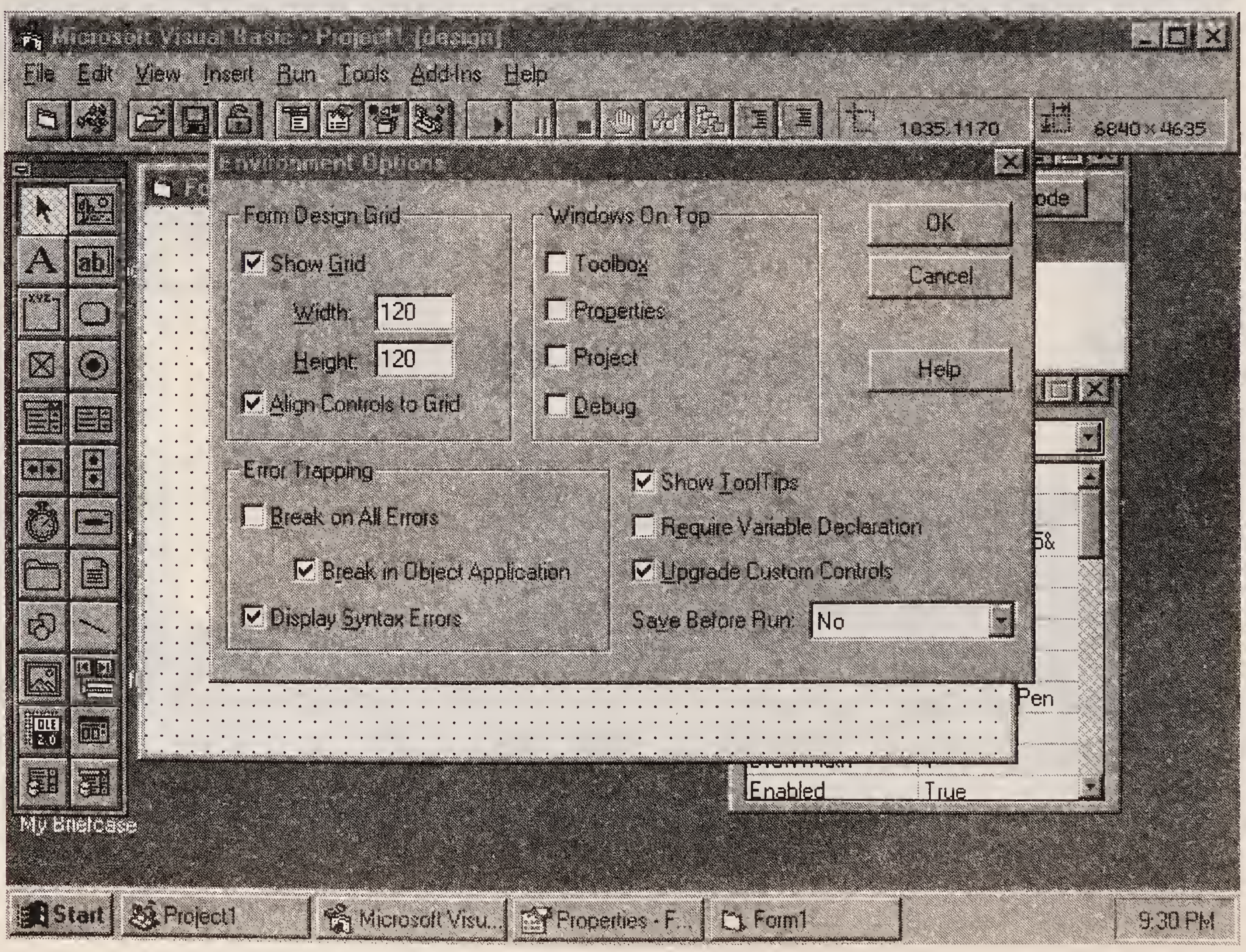
选择用户要存储的模块，单击 File 菜单的 Save File 命令，这时出现和图 5.26 相同的对话框，用户可在文件名文本中键入文件名。如果用户修改了已存在的文件再用该命令存储，将不再出现对话框。

5.6.4　用新的文件名存储窗体或类模块文件

单击 File 菜单的 Save As File... 命令，将出现如图 5.26 的对话框，该对话框允许将现有文件名改名（包括文件的路径名），这是在 Visual Basic 中做备份拷贝的唯一途径。

5.6.5　自动存储用户工作

Visual Basic 允许用户在运行一个工程文件时自动存储用户工作。单击 Tools 菜单的 Environment Options 命令，这时出现一对话框，见图 5.27。用户可以看到在对话框的左下角有一"Save Before Run"项，单击其右边的箭头打开列表框见图 5.28，选择其中的 Yes 项，然后单击 OK 按钮就可实现自动存储用户的工作。


Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
1035,1170
6840 x 4635
Environment Options
OK
Cancel
Help
Enabled True
Pen
My Briefcase
Start
Project1
Microsoft Visu...
Properties - F...
Form1
9:30 PM

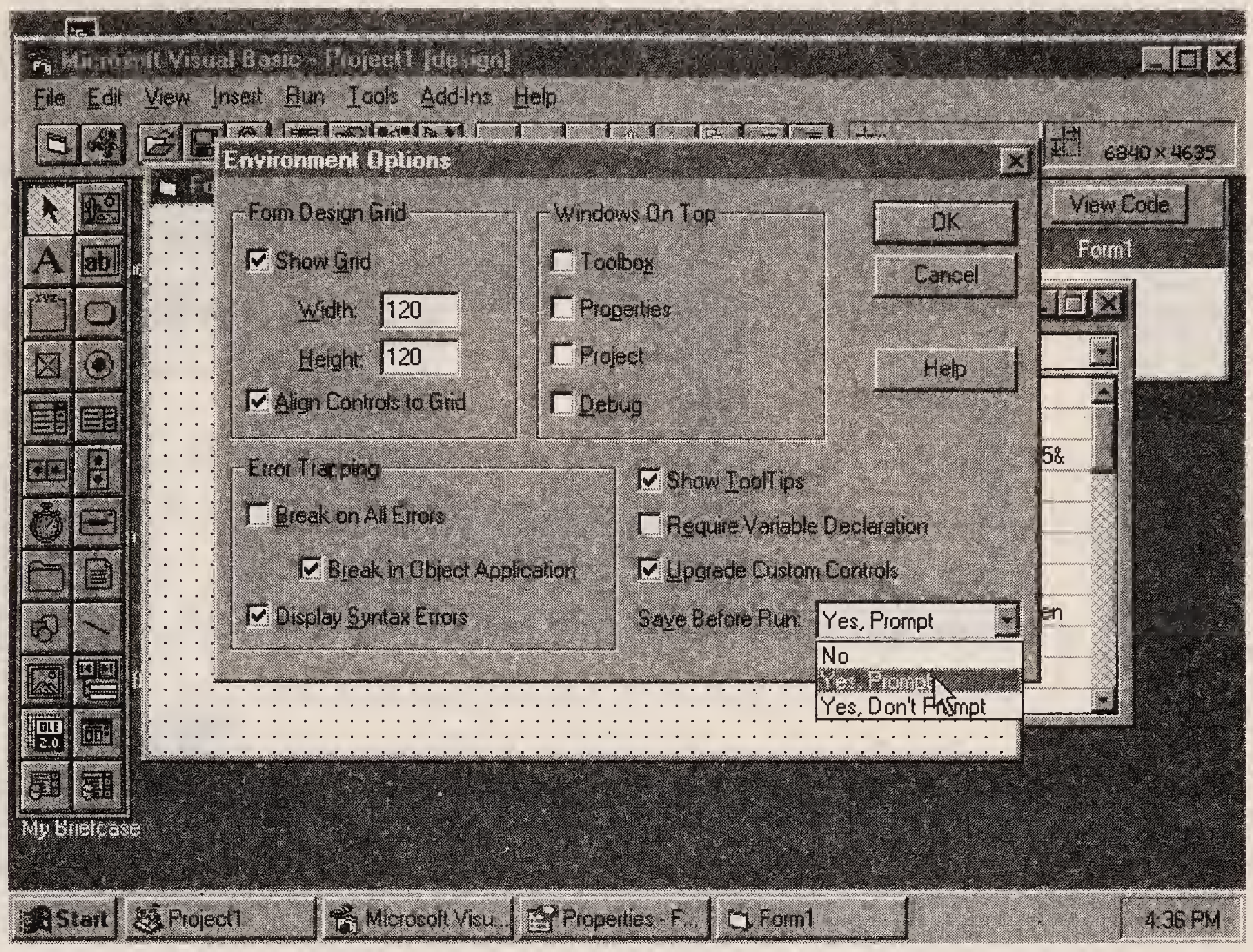
Form Design Grid
☑ Show Grid
Width: 120
Height: 120
☑ Align Controls to Grid

Windows On Top
☐ Toolbox
☐ Properties
☐ Project
☐ Debug

Error Trapping
☐ Break on All Errors
☑ Break in Object Application
☑ Display Syntax Errors

☑ Show ToolTips
☐ Require Variable Declaration
☑ Upgrade Custom Controls
Save Before Run: No

图 5.27　Environment Option 对话框


Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
6840 x 4635
OK
Cancel
Help
View Code
Form1
My Briefcase
Start
Project1
Microsoft Visu...
Properties - F...
Form1
4:36 PM

Environment Options

Form Design Grid

☑ Show Grid

Width: 120

Height: 120

☑ Align Controls to Grid

Windows On Top

☐ Toolbox

☐ Properties

☐ Project

☐ Debug

Error Trapping

☐ Break on All Errors

☑ Break in Object Application

☑ Display Syntax Errors

☑ Show ToolTips

☐ Require Variable Declaration

☑ Upgrade Custom Controls

Save Before Run: Yes, Prompt

No

Yes, Prompt

Yes, Don't Prompt

图 5.28 实现自动存储

5.7 建立 Windows 应用程序

Visual Basic 最大的优点能够方便地将用户的工程文件转化为独立的 Windows 应用程序，能够把工程文件的各部分连接成带有 EXE 扩展名的可执行文件。用户单击 File 菜单的 Make EXE File 命令将打开一对话框，见图 5.29。用户可在文件名文本框中键入文件名，其缺省名是工程文件名加上 EXE 扩展名，最后单击 OK 按钮即可生成可执行文件。

该执行文件可在非 Visual Basic 环境下运行。运行可执行文件，VBDLL400. DLL 的动态链接库必须在目录\Win95\SYSTEM 或用户路径的目录下。生成可执行文件的 Make EXE File 命令不影响工程文件。用户能够在不影响可执行的文件的情况下修改原始文件。

用户还可利用对话框中的 Options 按钮设置可执行文件相关的属性。单击 Options 按钮将出现图 5.30 的对话框。对话有以下几部分。

(1) Version Number

Major：主版本的工程文件号(0—9999)

Minor：次版本的工程文件号(0—9999)

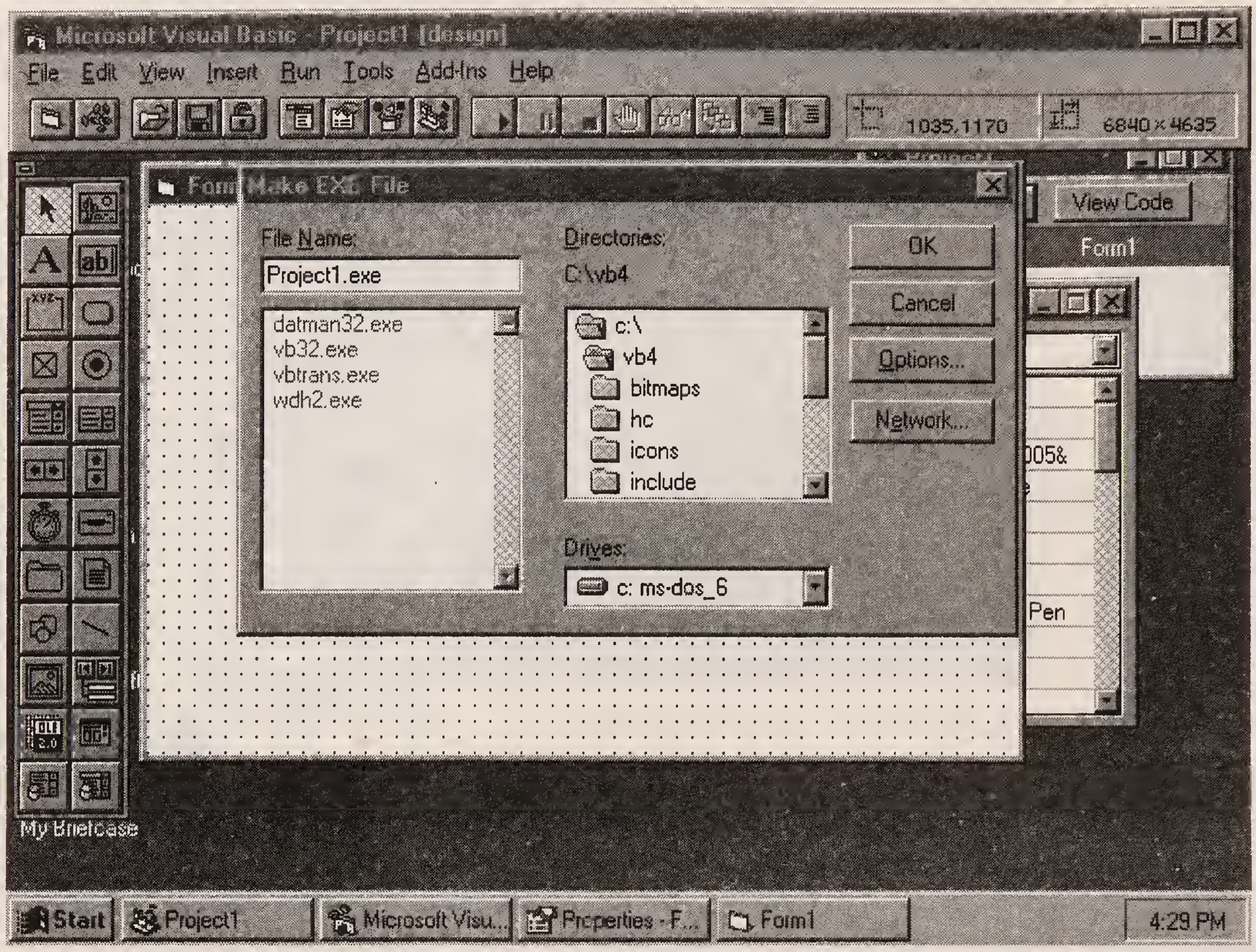
Revision：工程文件的修正版本号(0—9999)

Auto Increment：如果被选择，则用户每运行一次，文件版本号自动增加

(2) Application

Title：应用程序的标题名

Icon：应用程序的图标


Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
1035,1170
6840 x 4635
Make EXE File
File Name:
Directories:
OK
Cancel
Options...
Network...
Drives:
View Code
Form1
My Briefcase
Start
Project1
Microsoft Visu...
Properties - F...
Form1
4:29 PM

File Name: Project1.exe

datman32.exe
vb32.exe
vbtrans.exe
wdh2.exe

Directories: C:\vb4

c:\
vb4
bitmaps
hc
icons
include

Drives: c: ms-dos_6

图 5.29　Make EXE File 对话框

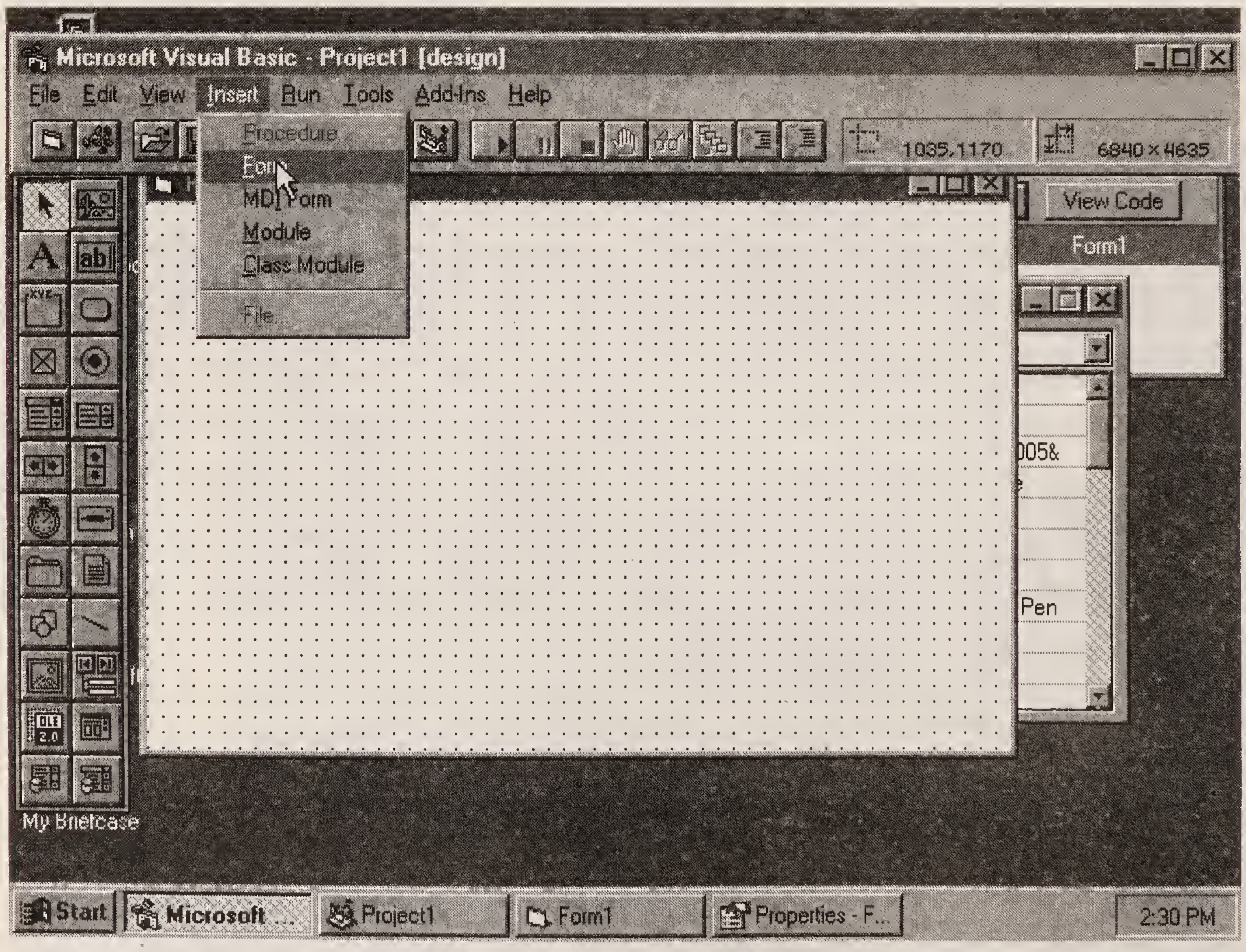

Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
Procedure
Form
MDI Form
Module
Class Module
File...
1035,1170
6840 x 4635
View Code
Form1
Pen
My Briefcase
Start
Microsoft ...
Project1
Form1
Properties - F...
2:30 PM

图 5.30　创建新窗体的菜单命令

(3) Version information

type:设置的版本信息类型,用户能够为应用程序增加诸如公司名称、文件描述和产品名称等信息。

value:为在类型框中被选择的信息类型键入相关信息。

5.8　多重窗体

到目前为止,用户已编写了几个小程序,但它们都含有一个窗体。在有些情况下,一个窗体能够满足用户程序相互交换信息的需要,但是在有些情况下,用户想增加更多的屏幕信息到一个应用程序中,以便应用程序为用户提供更为友好的人机对话界面。Visual Basic 4.0 允许用户增加一个或多个窗体到程序中去,每增加的一个新窗体都被认为是一个对象,并且都包含有它自己的对象、属性和事件过程,用户可以通过单击 Insert 菜单上的 Form 命令或工具条上的 Form 按钮来创建一个新窗体,见图 5.30。

在 Visual Basic 4.0 中,用户在运用窗体时有很大的灵活性。用户能够使在程序中的多个窗体同时可见,也可以根据需要用 Load 方法加载窗体或用 unload 方法释放窗体。如果用户在同一时刻显示了多个窗体,用户可以在它们之间进行切换或者控制窗体被使用的顺序。窗体有模态和非模态之分,模态化窗体一直保持焦点直到用户单击 OK 或 Cancel 按钮,非模态化窗体则相反。大多数基于 Windows 上的应用程序使用非模态化窗体,因为它给予用户更大的灵活性。当用户创建一个新窗体时,缺省窗体类型为非模态化窗体。

当用户创建一个新窗体,用户能够用 Load 方法把它加载到内存中,并通过在事件过程中用特定的语句来访问它,例如:

```
Load Form2
```

该语句将把程序中的第二个窗体加载到内存中。当用户想显示一个被加载到内存的窗体,可以通过用 Show 方法。Show 方法的语法格式为:

```
formname .show mode
```

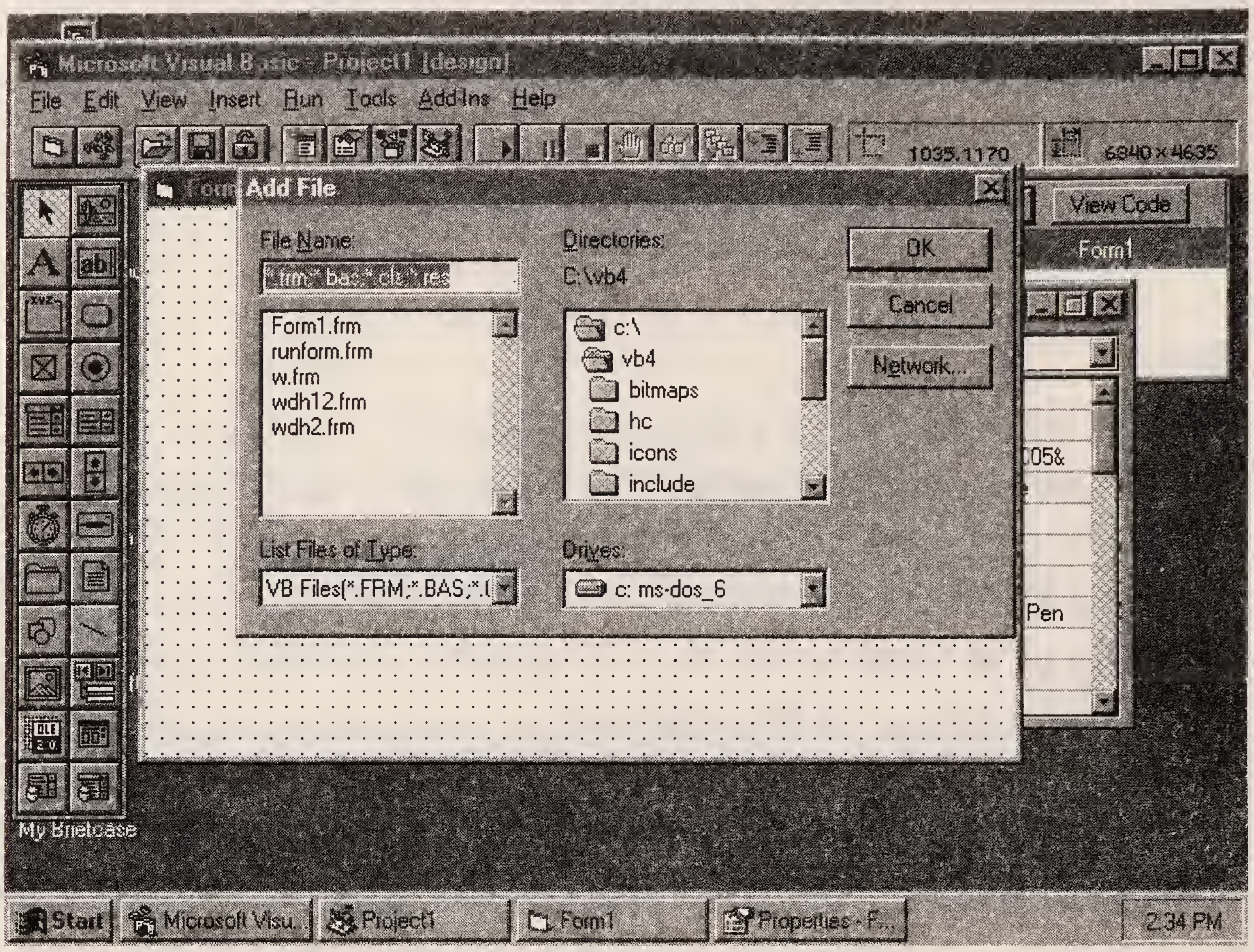
其中 formname 是窗体名,mode 为 0 代表非模块化窗体(缺省),mode 为 1 代表模态化窗体。例如:

```
Form2.Show
Form2.Show 1
```

如果用户在用 Load 方法之前用了 Show 方法,Visual Busic 将自动装载和显示窗体。

用户也可以用 Hide 方法隐藏一个窗体,也可以用 unload 方法从内存中释放被装载的窗体。Hide 方法虽然使窗体不可见,但是一直被保存在内存中以便用户在以后的程序中应用。

Visual Basic 允许用户在新的工程文件中重新使用已存在的窗体,这是因为用户以独立的.frm 文件保存了用户的窗体。为了增加一个已存在的窗体到新的工程文件中,用户可以单击 File 菜单下的 Add File... 命令,这时出现 Add File 对话框,见图 5.31。双击对话框中的窗体名,Visual Basic 将增加特定的窗体到工程文件中。


Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
1035,1170
6840 x 4635
Add File
OK
Cancel
Network...
View Code
Form1
My Briefcase
Start
Microsoft Visu...
Project1
Form1
Properties - F...
2:34 PM

File Name: *.frm;*.bas;*.cls;*.res

Form1.frm
runform.frm
w.frm
wdh12.frm
wdh2.frm

Directories: C:\vb4

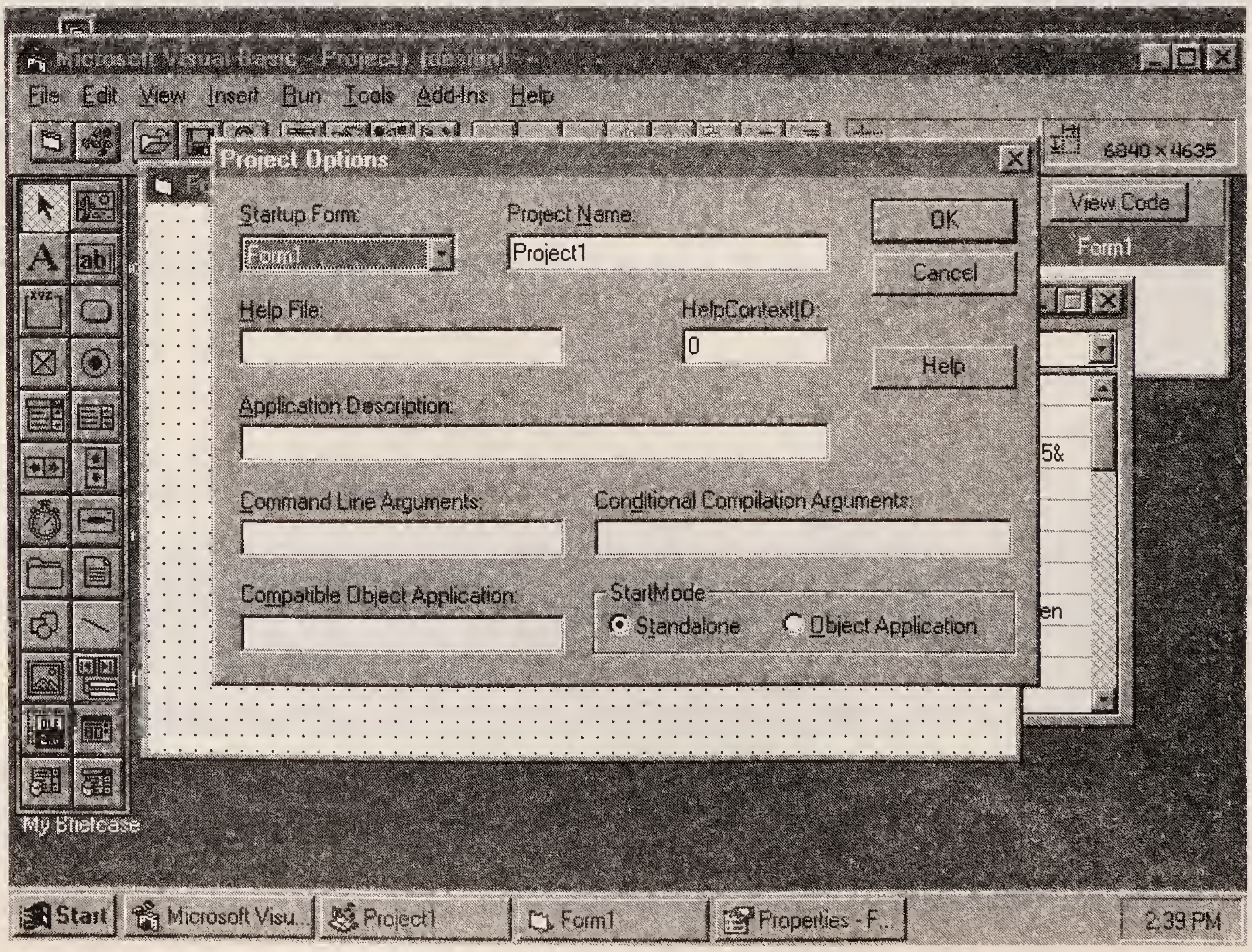
c:\
vb4
bitmaps
hc
icons
include

List Files of Type: VB Files(*.FRM;*.BAS;*.(

Drives: c: ms-dos_6

图 5.31　**Add File 对话框**

在拥有多个窗体的程序中，用户必须指定一个开始窗体。用户可以单击 Tools 菜单的 Project Options 命令，这时屏幕上弹出一对话框，见图 5.32。用户可以在对话框中“Start up-Form”的下拉式列表中选择一个窗体作为开始窗体。


Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
6840 x 4635
Project Options
OK
Cancel
Help
View Code
Form1
My Briefcase
Start
Microsoft Visu...
Project1
Form1
Properties - F...
2:39 PM

Startup Form: Form1

Project Name: Project1

Help File:

HelpContextID: 0

Application Description:

Command Line Arguments:

Conditional Compilation Arguments:

Compatible Object Application:

StartMode: ● Standalone ○ Object Application

图 5.32　**Project Options 对话框**

在大多数的程序中，用户可能建立一个主窗体完成他的大部分工作，然后添加特定目的的窗体用来处理程序的输入和输出。然而，Visual Basic 允许用户建立起一种等级制度，这种等级制度是在程序中窗体的一种特殊联系，这种特殊的联系使程序更好地运行。这些特殊的窗体称为 MDI（多文档界面）窗体，通过它们在程序中扮演的角色被区分为父窗体和子窗体。用户能够单击 Insert 菜单的 MDI Form 命令来创建一个 MDI 父窗体。用户也可以通过单击 Insert 菜单上的 Form 命令，然后设置窗体的 MDIChild 属性为 True 来创建一个 MDI 子窗体。在运行时，MDI 窗体像普通窗体一样工作，但也有以下特例：

（1）所有的子窗体被显示在父窗体的窗口边界内。

（2）当子窗体被最小化时，它不是在任务条上以一个按钮显示，而是在 MDI 窗体上缩小成一个小标题条。

（3）当父窗体被最小化时，父窗体和它所有的子窗体在任务条上显示为一个按钮。

（4）所有子窗体的菜单被显示在父窗体的菜单条上。当子窗体被最大化时，它的标题被显示在父窗体的标题条上。

（5）通过设置父窗体的 AutoShowChildren 属性，用户能够显示父窗体的所有子窗体。

在所谓的多文档应用程序中，父窗体和子窗体有着特别的应用。为了学习关于 MDI 窗体的更多知识和它相关的应用程序，请读者参看《Visual Basic 4.0 从入门到精通》的多文档界面应用程序一章。

第六章　基本语言

Visual Basic 作为一种语言，也有其控制结构、数据类型和语法等多项内容。为了能更好的使用这一语言，用户必须了解构成 Visual Basic 基础的程序语言。

假如用户对 Basic、Quick Basic 等已经非常熟悉的话，可以略过这一章内容。由于本书的重点是放在操作方面，而不是编程，所以本章只是对 Visual Basic 的一些基本语言做一简单介绍。

6.1　变　　量

Visual Basic 的指令都以一种方式处理计算机内存中的内容，甚至 Break 和 Stop 命令都修改计算机内存。但这些语句并不告诉用户它正在修改什么内容。因此，用户只能看到这些命令产生的结果。

像 Stop 这样的语句实际上是一种处理内存值的间接方法。用户想要直接处理内存值，就要把那些值表示为变量(Variable)。

变量就是用来保存信息和计算结果的描述名，是值和文本的逻辑容器。当在程序中使用变量时，实际上使用的是变量所指的内存中的内容。使用变量名使计算机程序增强了可读性。

6.1.1　声明变量

一个变量名的长度最长不超过 40 个字符，可以由字母、数字和下划线组成，但变量的第一个字符必须是字母，尽管变量名最长可达 40 个字符，而实际上变量名一般不应太长，否则会降低程序的可读性，只要能表达出这个变量名所能代表的变量特性就可以了。

在 Visul Basic 程序中，变量的声明是该变量第一次在源码中被引用时声明的。变量在程序中第一次使用时即为它声明的时刻，变量的声明就是在程序中用一合法值对之进行赋值的第一条语句。例如，下面的语句为最简单的声明：

```
a=1
```

在这里，a 为变量名，并把 1 赋值给它。这实际上是一个数学表达式。在这以后，只要程序中出现 a，Visual Basict 就自动给它赋值为 1，直到某些指令以明确的方式对其进行了修改。用户在建立变量时，最好不要只用一个字母，特别是在比较复杂的程序中。在这里，用一个字母作为变量名，只是为了方便。

6.1.2　变量类型

在 Visual Basic 语言中，有两种变量，数值型变量和字符串。下面先介绍一下数值类型。

1. 数值类型

Visual Basic 有六种数值类型。分别是 Integer(整型)、Long(长整型)、Single(单精度型)、Double(双精度型)、Variant(通用型)以及 Currency(货币型)。Integer 和 Long 数值类型用于保存整数,Single 和 Double 数值类型用于保存带小数的实数;Currency 是一种特殊的数值类型,主要用于计算十进制数,这种数值类型主要用于清除由二进制向十进制转换时造成的四舍五入错误。Variant 则使用赋值语句使用的数据类型来保存数值。例如,把一单精度(Single)实数赋给一个 Variant 型变量,则以后的操作即把这一变量作为 Single 型数值来对待。

(1) 整型数值

Integer 和 Long 型数值都是整数。用户在定义某一变量为整型变量后,给这一变量只能赋予整数。整型数的运算比浮点数的运算要快得多,所占的内存也较少。Interger 型的数值有一定的范围,－32768～32767,如果超出这一范围,就要用 Long 型。

(2) 浮点数值

单精度(Single)和双精度(Double)型数值即为浮点数值,它表示的是带小数的实数。单精度型能精确到七位,而双精度能精确到 15 位。用户在选用这些数据类型时,要注意变量所取值的范围。并且数值为整数时,为了运算速度,不用把变量硬定义为浮点型。对于大的浮点数,可用 D,E 格式输入输出。例如:

2.75E＋04,它表示 2.75×10^4 为单精度型

1.23E－12,它表示 1.23×10^{-12} 为单精度型

0.18D＋03,它表示 0.18×10^3 为双精度型

(3) 货币型

Currency 数据类型是定点数,用于计算货币的数量。由于处理的是货币,自然要求高的精度。而计算机在进行运算时,总是通过二进制数据进行运算,然后再根据用户的要求转换为八进制、十进制等数。因此,会导致进位误差。例如,有时我们计算 1＋1,有可能得到的结果却是 1.9999...。而 Currency 数值类型却是用十进制进行运算,因而其计算非常精确。

(4) 通用型

Variant 数值类型是对所有的未定义的变量的缺省数据类型。即当给某一变量赋值时,不管是什么类型的,Variant 都对其进行保留。要检测 Variant 变量中保存的数值究竟是何种类型,可用 VarType 函数进行检测,它的返回值表示的数据类型如下表所示。

Code	数据类型
0	空——未声明
1	空——已声明,但未包含数据
2	整型
3	长整型
4	单精度型
5	双精度型
6	Currency(货币)型
7	Date(日期)型

8	字符串型

2. 字符串

字符串定义一个文本字符序列。尽管字符串包含的大多是可打印字符，但也会包含一些非打印的控制字符，如回车符和换行符，也可包含一些无任何字符的空字符串。

Visual Basic 将文本赋值给字符串变量，如：

```
a$=″Q″
```

这条语句是把 Q 赋值给字符串变量 a$。为一个字符串变量赋值的简单的赋值表达式是一个等式，它使用如下语法：

```
address$=″Beijing″
```

美元符号 $ 用来标记字符串变量，即定义 address 为字符型变量。这是说明变量的一种方式，下一节我们将对这一种方式及另一种方式进行介绍。

一个字符串表达式是用引号括起来的字符串，也可以用连接操作符（+）把几个字符串连接起来，形成一个新的字符串。例如：

```
MyName$=″Zhang″
YourName$=″Wang″
OurName$=″Zhang″+″Wang″
```

这个例子就是将字符串“Zhang”、“Wang”合成一个大字符串“ZhangWang”，并赋值给字符串变量 OurName$。

6.1.3 变量说明

变量说明一般有两种方式：

（1）第一种方式

在使用的变量名后面加类型定义型。

例如：

Varl%	定义 Varl 变量为整型变量
Total#	定义 Total 变量为双精度型变量
Name$	表示 Name 为字符型变量

（2）第二种方式

在变量名前用关键字说明。经此说明以后，往后用到的变量名就不必在变量名后加类型定义符了。

1）Dim，在模块、窗体和过程中说明变量和数组。

说明格式

Dim 变量名称 As 类型名称

例如：

Dim	Number	As Integer	（定义 Number 为整型变量）
Dim	Address	As String	（定义 Address 为字符串）
Dim	Total	As Double	（定义 Total 为双精度变量）

2）Static 关键字

用在过程中定义静态变量及数组变量。用Static 说明的变量每当此过程被调用时，变量值都会继续保留。

说明格式

```
Static     变量名称[(下标)]    As              类型名称
```

例如：

```
Static     Data                As Interger     (定义 Data 为整型变量)
Static     Line(1,2)           As Single       (定义 Line(1,2)为单精度数组变量)
Static     YourName            As String       (定义 YourName 为字符串)
```

用 Static 和 Dim 定义的变量会有所不同，用 Dim 定义的变量每次调用时，都会被重新设置，(数值变量重新设置为 0，而字符串被设置为空串)。而用Static 定义的变量在每次被调用时，都将保留被调用前的值。

3) ReDim 关键字

用在函数中说明动态数组和重新分配数组空间，用来定义数组的大小。

说明格式：

```
ReDim      变量名称[(下标)]    As              类型名称
```

例如

```
Dim        Room()              As Integer      (定义 Room()为整型变量数组)
       .
       .
       .
ReDim      Room()                              (为 Room()数组定义大小)
```

4) Global 关键字

在全局模块(Global Module)中定义整型变量和数组。

说明格式：

```
Global     变量名称[(下标)]    As              类型名称
```

例如

```
Global     Total               As Integer      (定义 Total 为整型变量)
Global     Total(1,3)          As Integer      (定义 Total(1,3)为整型变量数组)
```

6.1.4　全局变量与局部变量

在对变量做了声明以后，在过程中就可以对变量进行访问了。但要注意，并不是所有的变量都是可以访问得到的，这和它们被定义的作用域有关。作用域就是用来描述程序在何处能访问到这些变量。在前面已讲过，对变量进行声明，可以用 Global、Dim 等。变量按其声明的不同，可分为全局变量和局部变量。

(1) 全局变量

用Global 声明的变量为全局变量，程序中的任何窗体和模块都可以访问到它。用Global 声明全局变量可以在任意模块或窗体中进行。用户在声明全局变量时要小心，注意到任何模块或窗体中对其进行修改，都将影响到其他模块中该变量的类型。

用 Global 声明全局变量，可以用一个例子来表示：

```
Global Team As String
```

不管这一声明是在程序的什么地方进行的，其他窗体或模块在访问该模块时，都将Team作一字符串变量。

（2）局部变量

在模块或窗体中用Dim或Static等声明的变量级别次于全局变量，只能在该模块或窗体中使用，其他模块等无权使用。这种变量叫做局部变量。例如：

```
Dim Name As String
```

这一声明是在某一模块或窗体中使用的，声明Name为字符串，别的模块或窗体要使用Name这一变量，得重新声明。

在单个过程中定义的局部变量级别最低，而且只能在声明这些变量的过程中使用，在别处不能访问这一变量。当过程结束时，局部变量的生命周期也结束了。当下次再调用这一过程时，上次过程中使用的变量将重新被赋值。要想重新调用这一过程并使用原来的变量值得在声明时用Static声明。

```
Static Name As String
```

全局变量级别最高，并且可以在任何模块或窗体中进行声明。因此，用户在声明全局变量时要注意，是否已存在这一变量。如果在声明局部变量时，变量名与全局变量名相同，则在这一模块中该变量是一局部变量，而在其他模块或窗体中该变量仍为全局变量。

6.2　基本语句

随着对Visual Basic的不断熟悉，读者会更加轻松自如地使用Visual Basic的指令和各种引用设置的方法。语句是引用指令集的非正式方法，且Visual Basic程序中的大多数源码都可以分成语句。语句是一组指令，这组指令的执行取决于数学表达式内变量的值。

6.2.1　条件语句

条件语句是测试一数值表达式，如果条件成立，则执行某一语句，否则执行别的语句。

（1）If/Then语句

格式为：

```
If<条件> Then<语句>
End If
```

If/Then语句测试一个数学表达式，查看该表达式的值是否为真（True）。若为真，则继续执行表达式之后的第一条相关指令。如果表达式的值为假（False），则跳出嵌套在语句中的指令。

例如：

```
If fingers=5 Then Text1. Text="My name is Wang"
End If
```

这一语句意思是如果fingers=5，则在文本框1中输出"My name is Wang"。否则就跳出这一语句。

（2）If/Then/Else语句

格式为：

```
If <条件1> Then <语句1>
Else If<条件2> Then <语句2>
.
.
.
Else If <条件n> Then <语句n>
End If
```

这个语句是查看一数学表达式，如果满足条件1，就执行语句1的指令，再查看是否满足条件2，如满足，就执行语句2的命令，一直检查到条件n。

(3) Select Case 语句

格式为：

```
Select Case <变量或(表达式)>
  Case<变量值1>
      <语句1>
  Case<变量值2>
      <语句2>
  .
  .
  .
  Case<变量值n>
      <语句n>
  End Select
```

Select Case 语句比较起始表达式和其他表达式的值。语句中含有多个比较表达式，但它们只和起始表达式相比较。起始表达式可由一个变量或表达式组成。对于每个比较表达式，只要其值为True，则执行该表达式之后的一组指令。当执行到下一个Case 或End Select 时，指令的执行则跳转到End Select 后续的指令。

例如，一个字符串比较的例子，Select Case 语句根据字符串变量的内容来输入不同的名字。

```
Select Case Name $
   Case "Your Name"
       Name1. Text="Wang Lin"
  Case "My Name"
       Name2. Text="Li Lin"
  case "His Name"
       Name3. Text="Feng Yi"
End Select
```

这段程序是根据不同的Name $ 中的内容，输出不同的姓名。

6.2.2 循环语句

反复执行一组指令的语句被称为循环语句。循环语句的每一次执行，称为一轮循环。FOR-NEXT 语句是 BASIC 最主要的循环控制语句，其他还有 Do 循环语句。循环是靠逻辑条件来结束的。循环语句起始与末尾之间的指令被往复执行，直到满足某些条件为止。如果循环是无条件的，指令的往复执行将永无止尽，直到用户强行中止循环。

(1) For 循环

格式：

```
For <变量>=初始值 TO 终止值 [Step 步长]
    [Exit For]
    语句
Next <变量>
```

For 循环语句用于封装重复执行的一组指令。首先变量被置初始值，然后按序执行循环体内的指令。每执行一个指令，则变量按步长增加，直到大于等于终止值为止，或者遇到执行 Exit For 语句时退出循环，执行 Next<变量>后续的指令。

格式中的(Step 步长)可有可无，若没有时，程序执行到 Next<变量>后，变量自动加 1。若有这一语句，则程序执行到 Next<变量>时，变量增加的值即为定义的步长。

For 循环中还可以嵌套一重或多重 For 循环，例如：

下面的程序是打印一个九九表：

```
Private Sub Form _ Click()
  AutoRedraw=True
  For a=1 To 9
    For b=1 To a
        Print a; " * "; b; "="; a * b; " ";
        If b=a Then Print Chr $ (13)
    Next b
  Next a
End sub
```

在这段代码中 AutoRedraw=True，表示当 Form 窗口较小时，一部分结果由于窗口边界的限制而不能显示出来，这时放大窗口，被遮盖的部分将显现出来。如果 AutoRedraw 值为 False，则当放大窗口时，遮盖的部分就显现不出来了。Char $ (13)表示回车。这样当运行程序时，单击 Form 窗口，即显示一标准的九九乘法表。

(2) Do 循环

格式 1 为：

```
Do [While|Until] <条件>
    <语句 1>
[Exit Do]
    <语句 2>
    ⋮
    <语句 n>
    Loop
```

格式 2 为：

```
    Do
        <语句 1>
    [Exit Do]
        <语句 2>
          ⋮
        <语句 n>
    Loop [While|Until] <条件>
```

Do 循环语句重复执行它包含的一组指令。指令的执行可能取决于可选的条件表达式，也可能不取决于条件表达式。如果条件满足，指令的执行可以终止。如果没有条件，循环则不会终止，只能用 Exit Do 语句来退出。

格式 1，条件表达式放在 Do 的后面。如果条件表达式前面含有关键字 While，只要表达式求值为 True，语句中的指令则按序执行。如果条件表达式前面含有关键字 Until，则当表达式求值为 False 时，执行语句中的指令。否则，如果 While 表达式为 False 或 Until 表达式为 True 时，程序则跳转到 Loop 后继续执行，或由 Exit Do 指令跳出循环。

格式 2，条件表达式位于语句的尾部，在 Loop 之后，While 和 Until 的用法与格式 1 中的用法相同。只是，语句中的指令在第一轮循环中是无条件执行的，当执行到 Loop 后，条件表达式才开始求值。

例：在程序运行时弹出一对话框，要求用户输入口令，如输入不正确，则继续出现对话框，直到输入正确口令为止。程序代码是：

```
  Private Sub Form _ Click()
    Do Until Pass $ ="LiMing"
    Pass $ =InputBox $ ("Please enter the password!")
    Loop
  End Sub
```

在这里需注意，只有在对话框中输入的口令完全正确时才会终止这一循环，即使写成“Liming”即该大写的地方写成了小写也不行。

6.2.3 转移语句

在以前的 Basic 语言中，每一行程序都有一个行号。要想实现转移，可以使用 GOTO 命令和 GOSUB 命令。这些命令后一般加的是行号。当程序执行到这些命令时，将暂停当前命令的执行，转而去执行指定点处的命令行，执行完毕后，再回到暂停的地方，继续执行下面的命令。

在 Visual Basic 中，行号已被标号代替。标号出现在一行内，后面跟一个冒号，标号标记一个转移点，用于为转移语句提供跳转点。

事实上，现在许多功能完善的现代 BASIC 程序中的源代码已经不再使用这两种语句。主要原因是不用这两种语句，可以编写出程序流程完美的应用程序。可在 Visual Basic 中，仍然保留有 GoTo 和 GoSub 语句，只是它们的作用已大大减弱了。用户可以在程序中选用这两

种语句,也可以完全不使用它们。

(1) GoTo 语句

格式:

GoTo ＜行标号＞

当程序执行到这条语句时,将暂停当前命令的执行,去执行行标号所在的那行语句。

(2) GoSub 语句

格式:

GoSub ＜行标号＞
＜行标号＞ 语句
.
.
.
Return

GoSub 语句在执行这一语句命令时,解释程序立即把指令的执行转移到行标号指定的指令。当子程序按序执行指令时,解释程序跟踪指令执行的位置。子程序执行到 Return 指令时,解释程序把控制返回给调用 GoSub 指令的下一条指令。

由于这两个语句不常使用,我们在这里就不再举例了。

6.3 常用函数

在 Visual Basic 中,有许多函数,可以对字符串、变量值等进行操作,下面,只对一些常用的函数做一简单介绍。

6.3.1 字符串函数

Visual Basic 有许多处理字符串的函数,这些函数包括:在字符串中查找字符串中某一段的函数,从字符串中提取某一段的函数,等等。

(1) LTrim $ () 用于去掉字符串左边的空白部分。

格式为:

LTrim $ (字符串)

例如:A $ ="xxx How are you"(x 表示空格)
B $ =LTrim $ (A $)

则 B $ ="How are you",A 字符串左边的空白部分被去掉了。

(2) RTrim $ () 用于去掉一个字符串的右边的空白部分。

格式为:

RTrim $ (字符串)

例如:A $ ="How are you xxxx"(x 表示空格)
B $ =RTrim $ (A $)

则 B $ ="How are you",A 字符串右边的空白部分被去掉了。

(3) Left $ () 用于取出字符串左边的某些字符。

格式为：

Left $（字符串，个数）

例如：A $ ="How are you"

B $ =Left $（A $ ，3）

则B $ ="How"，取出了 A 字符串的前三个字符。如果要取的个数大于原字符串中的字符个数，则与原字符串相同，即只是将原字符串中的全部字符取出而已。

（4）Right $（）用于取出字符串中右边的一些字符。

格式为：

Right $（字符串，个数）

例如：A $ ="How are you"

B $ =Right $（A $ ，3）

则B $ ="you"，取出了 A 字符串中右边的三个字符。如果原字符串为空串，即没有字符，则新的字符串也为空串。如果要取的字符个数大于原字符串的个数，则新字符串与原字符串相同。

（5）Mid $（）用于取出字符串中任意字符。

格式为：

Mid $（字符串，起始位置，个数）

例如A $ ="How are you"

B=Mid $（A $ ，5，3）

则 B $ ="are"，取出了 A 字符串中第五个字符开始的三个字符 。

（6）Len（）用于计算某字符串的长度或某变量所占字节数(Bytes)。

格式为：

Len（字符串）

Len（变量）

例如：A $ ="How are you"

B=Len $（A $ ）

则 B=11，因为 A 字符串有 11 个字符。

例如：A=Len（变量%）

A=2，因为%表示为整型变量，占 2 个字节数。

（7）Ucase $（）用于将字符的小写字母转换为大写字母。

格式为：

Ucase $（字符串）

例如：A $ ="How are you"

B $ =Ucase $（A $ ）

则 B $ ="HOW ARE YOU"

（8）Lcase $（）用于将字符串的大写字母转换为小写字母。

格式为：

Lcase $（字符串）

例如：A $ ="How are you"

B$=Lcase$(A$)

则 B$="how are you"

(9) Space$()用于返回指定数目的空格字符串。

格式为：

Space$(个数)

例如：A$=Space$(3)

则 A$="xxx"(在这里，x 表示空格)

(10) String$() 返回以 ASCII 码或某一字符串的第一个字符串表示的字符串。

格式为：

String$(个数，ASCII 码)

String$(个数，字符串)

例如：A$=String(3,69)

B$=String(3,"Good")

则A$="EEE"，因为 E 字符的 ASCII 码为 69。

B$="GGG"，因为 G 为"Good"字符串的第一个字符。

6.3.2 转换函数

(1) Chr$()与 Asc()

Chr$是用来将 ASCII 码转换成字符，而 Asc 则将字符转换为 ASCII 码。

格式为：

Chr$(数值表达式)

Asc(字符)

例如：A$=Chr$(69)

B=Asc("E")

则A$="E"　　因为 ASCII 码为 69 代表字符“E”。

B=69　　E 的 ASCII 码为 69。

(2) Str$()与 Val()

Str$将数值转换为字符串类型，而 Val 则将字符串类型的数值转换为数值：

格式为：

Str$(数值表达式)

Val(字符串类型的数值表达式)

例如a%=123

A$=Str$(c%)="123"，将数值 123 作为字符串输出。

B=Val(A$)=123，将字符串 123 作为数值输出。

(3) CDbl(x)

将参数 x 值转换为双精度型数值。

(4) CLong(x)

将参数 x 值转换为长整型数值。

(5) Csng(x)

将参数 x 值转换为单精度型数值。

(6) Fix(x)

将参数 x 值的小数部分舍去，转化为整数输出。

(7) Int(x)

将参数 x 值转换为小于等于该参数的最大整数输出。

在这里需注意，在对负数进行整数转换时，用 Fix(x)与 Int(x)函数结果会有所不同。

例如Int(－3,2)＝－4，因为－4 为小于等于－3.2 的最大整数。

Fix(－3,2)＝－3，因为 Fix 函数只是将小数部分舍去。

(8) Ccur(x)

用于将 x 值转换为货币值(Currency Value)。小数部分最多取 4 位。

例如x＝789.345721

y＝Ccur(x)

则 y＝789.3457

(9) Format $ ()

Format $ ()函数也可以将数值转化为字符串类型，但它能控制输出字符串的格式。例如可以在输出字符串前加符号 $ 等。

其格式为：

Format $ (数值表达式，fmt $)

其中 fmt $ 字符串是由一些符号组成的表示所需要数值的格式。如果省略 fmt $ ，则 Format $ ()函数与 Str $ ()函数的功能相同。

例如：Format $ (0.565,"0.00%")

Format $ (765432.1,"###,# 0.00")

则输出分别为

```
56.50%
765,432.10
```

6.3.3 数学函数

(1) Sin(x)

返回 x 的正弦值，在这里 x 的单位为弧度。

(2) Cos(x)

返回 x 的余弦值，x 的单位为弧度。

(3) Tan(x)

返回 x 的正切值，x 的单位为弧度。即 tg(x)。

(4) Atn(x)

返回 x 的反正切值，x 的单位为弧度。即 arctg(x)。

(5) Log(x)

返回参数 x 的自然对数。

(6) Exp(x)

返回以 e 为底 x 的指数值。即 e^x。

(7) Sqr(x)

返回参数 x 的平方根值，即$\sqrt{x}$。

(8) Abs(x)

返回参数 x 的绝对值。即|x|。

(9) Hex $ (x)

返回参数 x 的十六进制值。

(10) Oct $ (x)

返回参数 x 的八进制值。

(11) Sgn(x)

返回参数 x 的符号，当 x>0 时返回 0，x<0 时返回－1。

(12) Rnd(x)

用于返回一个介于 0～12 之间的单精度随机数。

6.3.4 日期函数

(1) Date $ 函数

用以返回计算机系统内的最新日期，此日期以“mm-dd-yyyy”格式显示，例如“11-28-1995”。表示 1995 年 11 月 28 日。

(2) DateSerial()函数

返回一个序数，代表该整型日期参数的日期，其参数值为整型。此序数为以“1899，12，30”为 0，向前为负，向后为正。日期介于 1753 年 1 月 1 日至 2078 年 12 月 31 日之间。

格式为：

DateSerial(Year,Month,Date)

例如：

DateSerial(1899,12,30)返回值为 0

DateSerial(1899,12,31)返回值为 1

DateSerial(1900,12,30)返回值为 365

(3) DateValue()函数

返回一个序数代表该日期字符串参数的日期。其功能与 DateSerial 几乎完全相同，只是 DateSerial()函数返回的参数值为整数，而 DateValue 返回的参数值却是字符串。

格式为：

DateValue(日期字符串)

例如：

DateValue(12-30-1899)返回值为 0

DateValue(12-31-1899)返回值为 1

DateValue(12-29-1899)返回值为－1

(4) Day()函数

返回日期代号(1～31)。

格式为：

Day(序数)

(5) Month()函数

返回月份代号(介于1～12)。

格式为:

Month(序数)

(6) Now 函数

返回系统内最新的日期和时间,但是以序数方式返回。

格式为:

Now

例如:

Format $ (Now,″mm-dd-yyyy″)

返回10-11-1995。

(7) TimeValue()函数

返回一个序数代表该时间字符串参数。序数介于0～0.999988425925926。时间字符串参数介于0:00:00～23:59:59之间。

格式为:

TimeValue(时间字符串)

例如:

TimeValue(″12:00:00″)返回0.5。

TimeValue(″00:00:00″)返回0。

(8) TimeSerial()函数

返回一个序数代表该整数时间参数的时间。参数值为整型。

格式为:

TimeSerial(小时、分、秒)

例如:

TimeSerial(″12:0:0″)返回0.5。

TimerSerial(″0:0:0″)返回0。

(9) Time $ 函数

返回计算机系统内的最新时间,此时间以“hh:mm:ss”格式显示。

格式为:

Time $

(10) Year()函数

返回年代(介于1753～2078之间)

格式为:

Year(序数)

例如:

Year(0)　　返回1899。

Year(365)　返回1900。

(11) Month()函数

返回月份代号(介于1～12之间)。

格式为：

Month(序数)

(12) Weekday()函数

返回星期代号(1 代表星期日，……，7 代表星期六)。

格式为：

Weekday(序数)

(13) Hour(),Minute(),Second()函数

分别返回小时(0～23)，分钟(0～59)，秒(0～59)。

格式为：

Hour(序数)
Minute(序数)
Second(序数)

6.4 数组的使用

数组并不是一种数据类型，而是一组相同类型的变量，可以当作一个对象来引用。数组变量必须用 Global 或 Dim 语句定义，语法格式如下：

```
Dim ArrayName([lower To] upper[,[lower To]upper])[As type]
```

在这里，每对 lower 和 upper 为数组的维数，lower 是索引值的低限，upper 是索引值的高限。如果省略 lower，则其缺省值为 0。数组的范围与整数的范围是相同的(－32768 到 32767)。单个数组的最大维数是 60。

例如：

```
Dim Cx(5)
Dim Cy(2,6 To 9)
```

第一个例子将 Cx 定义为一维数组，它具有 5 个元素，可以索引值放在括号里访问数组中的每个文本。数组的索引值就指出了要使用的数组元素。例如 Cx 的元素为：

```
Cx(0)
Cx(1)
Cx(2)
Cx(3)
Cx(4)
```

第二个例子中定义了两维数组 Cy(2,6 To 9)，二维数组中第一个数为行索引值，第二个元素为列索引值，数组元素为：

```
Cy(0,6)  Cy(0,7)  Cy(0,8)
Cy(1,6)  Cy(1,7)  Cy(1,8)
```

数组用于保存大块的数据。用户可以通过不同的下标来保存一组数据中的各个数据。

6.5 数字表达式的计算

Visual Basic 中的数字表达式同一般的算术式比较类似。也就是说，数字表达式的赋值语句同字符串的赋值语句差不多。

6.5.1 算术运算符

表 6.1 中列出 Visual Basic 中的算术运算符，包括标准的加、减、乘、除以及乘方、取模运算和整数。运算优先级表示当表达式包含多个操作符时，先执行哪个操作符。

表 6.1 Visual Basic 算术运算符和优先级

运算符	运算	优先组
+	加	6
−	减	6
−	取负	2
*	乘	3
/	除	3
\	整除	4
^	乘方	1
Mod	取模	5

6.5.2 关系运算符

关系运算符如表 6.2 所示，用来比较两个数并返回逻辑值。如果两个数的关系与关系运算符一致，则表达式返回值为 True(−1)，否则返回 False(0)。

表 6.2 Visual Basic 关系运算符

运算符	运算
=	等于
>	大于
>=	大于等于
<	小于
<=	小于等于
<>	不等于

6.5.3 逻辑运算符

逻辑运行符如表 6.3 所示。对两个逻辑值运算，并返回一个逻辑值。另外，如果不是对逻辑值运算，而是对数字运算，则以数字逐位进行逻辑运算。所有的逻辑运算具有相同的优先级。逻辑运算只涉及到两个值：True(−1)和 False(0)。

表 6.3 Visual Basic 的逻辑运算符

运算符	运算
Not	取反
And	逻辑与
Or	逻辑或
Xor	逻辑异或
Eqv	逻辑相符
Imp	逻辑隐含

6.6　赋值语句和属性设置

给变量赋值和设定属性是 Visual Basic 代码中两个常见的任务。Visual Basic 使用等号（=）来完成上述两个操作。例如：

```
Rate=0.1
```

左侧的 Rate 为变量名，右边 0.1 则为赋给 Rate 的值。这样，在这一代码过程中只要出现 Rate，它所代表的值即为 0.1。

用户在选定某一对象后，总要为其设置属性。通常是通过对其 Properties（属性）窗口中的属性值直接进行设置来完成。当用户想要在程序运行时对属性进行设置或某一对象的属性值必须在程序运行时不断的进行变化的时候，就必须通过代码来实现。其格式为：

```
Object.Property=Value
```

Object 为用户选中对象的控制名。Property 为想要改变设置的属性，如 BackColor，Visible 等。Value 为设置的属性值。

例如，假设有一文本框，其控制名为 Text1，我们想要改变其 Text（文本）值为空，即在文本框中输出空字符串。为此，只要在一事件过程中加入下一行：

```
Text1.Text=" "
```

则 Visual Basic 在执行这一语句时，即在 Text1 文本框中输出一空字符串。如果将其改为：

```
Text1.Text="Welcome to Visual Basic!"
```

则 Visual Basic 在执行这一语句时，在 Text 文本框中输出

```
Welcome to Visual Basic!
```

用户可以通过这一方法来对不同的对象在程序运行时进行属性设置。

下面举一个例子，在这个例子中，通过对窗体 BackColor 属性进行设置，在每次单击 Form 窗口时，Form 窗口将显示不同的颜色。

（1）选择 File 菜单中的 New Project 命令，将 Form 窗口的 Caption 属性设置为 ColorTest。

（2）双击 Form 窗口，或单击 Project 窗口中的 View Code 按钮，打开 Code 窗口。

（3）选择 Object 列表框中的 Form 与 Procedure 列表框中的 Click，在 Form _ Click()事件的过程头与过程尾之间输入下列代码：

```
Private Sub Form _ Click()
  Form1.BackColor=QBColor(Int(Rnd * 15))
End Sub
```

这段代码的意思是每次单击 Form 窗口时，将改变其背景颜色。由于 Rnd 函数输出一 0 到 1 之间的随机数值，所以每次执行这一段代码时，由于 Rnd 不断变化，因而 QBColor()函数值不断变化，引起 Form 窗口内 BackColor 属性随之改变，因而每次单击 Form 窗口时，其背景总显现一种新的颜色。

（4）关闭 Code 窗口，单击工具栏中的 Start 按钮，在 Form 运行窗口单击鼠标左钮，每单击一次，出现一种颜色。

6.7 修改 myfirst 工程文件

在前面第四章中的 myfirst 工程文件例子中。还有一些不完善的地方，那是因为我们还没有介绍 Visual Basic 中的各个控件的作用、属性以及 Visual Basic 的基本语言。既然现在已对这些内容做了简单的介绍，下面就对 myfirst 工程文件进行修改，使其完善一些。

(1) 单击 File 菜单中的 Open Project 命令，在 Open Project 对话框中选择 C:\vb 路径下的 myfirst 工程文件。单击 OK 按钮，这时如果当前的工程文件并未存盘或者已存盘但做了修改了，则 Visual Basic 即会弹出一对话框，询问用户是否将文件或修改后内容进行存盘。用户可以通过单击 Yes 或 No 按钮来进行选择。

(2) 当 Visual 将当前文件存盘或取消后，即装入 myfirst 工程文件，这时的 Form 窗口并不可见。用户可以单击 Project 窗口中的 View Form 按钮来显现它。

(3) 单击 Project 窗口中的 View Code 按钮，打开 Code 窗口。

(4) 选择 Ojbect 列表框中的 data1 与 Procedure 列表框中的 GotFocus，在 data1 _ GotFocus()事件过程头与过程尾间键入下列代码：

```
Private Sub data1 _ GotFocus()
data1.SelStart=0
data1.SelLength=Len(data1.Text)
End Sub
```

这段代码的意思是当 data1 文本框获得焦点时，其中原有的内容将被一高亮所覆盖，且输入的新内容将覆盖原有的内容。

(5) 选择 Object 列表框中的 data2 与 Procedure 列表框中的 GotFocus，在 data2 _ GotFocus()事件过程头与过程尾间键入下列代码：

```
private Sub data2 _ GotFocus()
data2.SelStart=0
data2.SelLength=Len(data2.Text)
End Sub
```

这段代码与(4)中事件过程代码的意思是一样的。

(6) 选择 Object 列表框中的 data1 与 Procedue 列表框中的 LostFocus，在 data1 _ LostFocus()事件过程头与过程尾间输入下列代码：

```
Private Sub data1 _ LostFocus()
  Dim I As Integer
  For I=1 To Len(data1.Text)
    A$ =Mid$(data1.Text,I,1)
    If Asc(A$)>=58 Or Asc(A$)<=47 And Asc(A$)<> 46 Then
    MsgBox("You should enter the digit!")
    data1.SetFocus
    Call data1 _ GotFocus
    Exit Sub
    End If
  Next I
End Sub
```

这段代码的意思是，当文本框 data1 失去焦点时，Visual Basic 依次循环将 data1 文本框中的字符转换为 ASCII 码，然后判断其是否是数字或小数点字符。当判断到某一字符不是数字或小数点时，弹出一对话框，提醒用户应输入数字。当单击 OK 按钮后，Visual Basic 将焦点设在 data1 文本框中，然后调用 data1 _ GotFocus 事件过程，即将 data1 文本框中的内容用高亮覆盖，等待用户重新输入，并退出 data1 _ LostFocus 事件过程。

(7) 选择 Object 列表框中的 data2 与 Procedure 列表框中的 LostFocus，在 data2 _ LostFocus()事件过程头与过程尾间输入下列代码：

```
private Sub data2 _ LostFocus()
Dim I As Integer
For I=1 To Len(data2. Text)
    B$ =Mid$ (data2. Text,I,1)
    If Asc(B$)>=58 Or Asc(B$)<=47 And Asc(B$)<> 46 Then
    MsgBox("You should enter the digit!")
    data2. SetFocus
    Call data2 _ GotFocus
    Exit Sub
    End If
  Next I
End Sub
```

这段代码与(6)中 data1 _ LostFocus 事件过程代码意思是相同的。

(8) 将原文件中的 add _ Click()事件过程中的那行代码。

```
data3. Text=Str$ (Val(data1. Text)+Val(data2. Text)
```

删除，并输入下列代码：

```
Private Sub add _ Click()
  Dimx,y,zAs Single
    x=val(data1. Text)
    y=val(data2. Text)
    z=x+y
  Data3. Text=Str$ (z)
End Sub
```

这段代码的意思是将 data1文本框中的内容赋值给 x，将 data2文本框中的内容赋值给 y，z=x+y，然后将 x 与 y 的和 z 用字符串形式在文本框 data3中输出。

(9) 关闭 Code 窗口，回到 Form 窗口，将 data1与 data2文本框的 Text 属性值设置为0，这样刚开始运行程序时，在 data1与 data2中会有缺省值0出现。

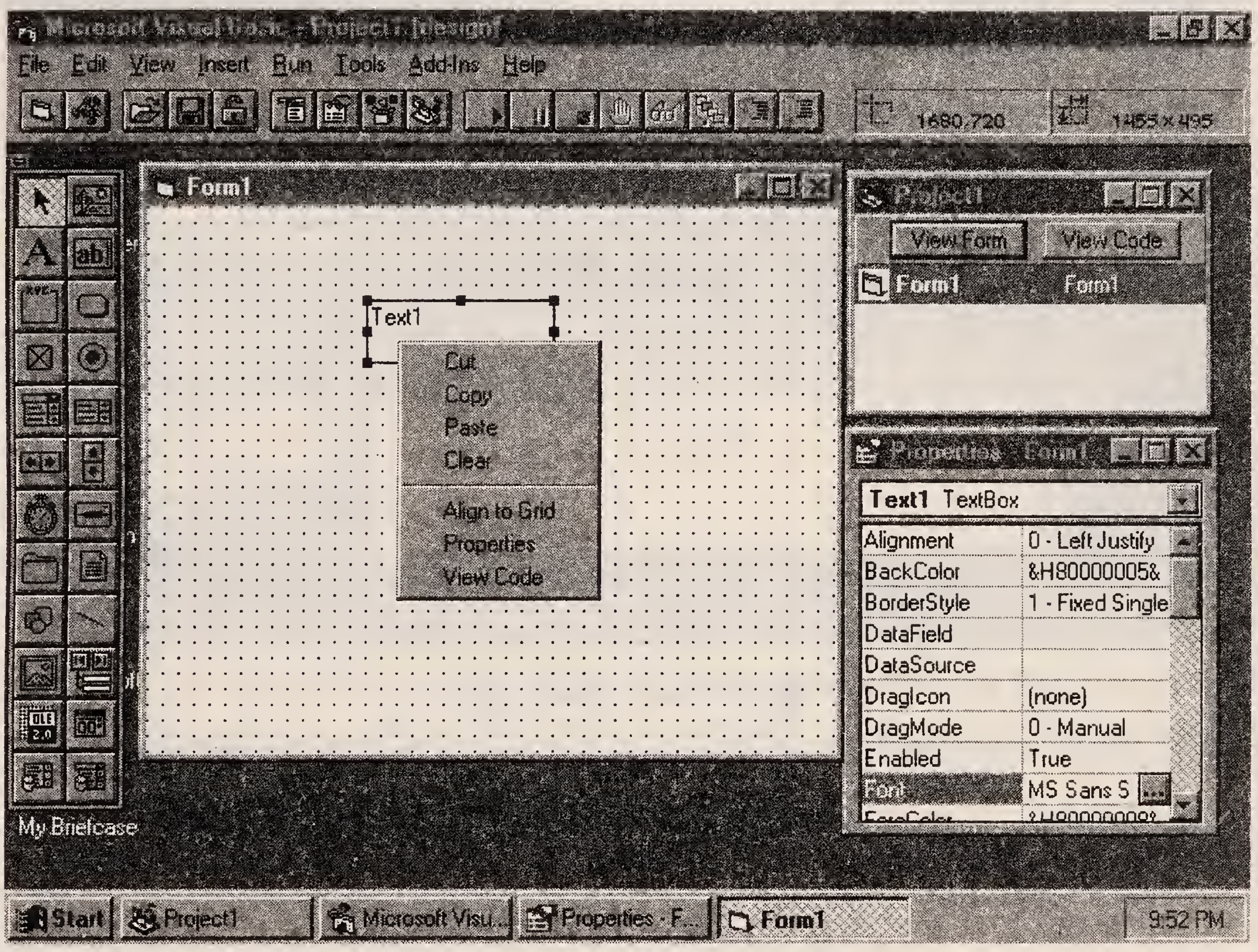
(10) 选择 File 菜单中的 Save File 与 Save Project 命令，将改动后的文件存盘。

(11) 单击工具栏中的 Start 按钮，运行程序，这时 data1文本框中的0将被一高亮所覆盖，输入数据则取代0。单击 Tab 键进入 data2文本框，输入一数据。单击 Add 按钮，即在 data3文本框中输出结果。如果在 data1与 data2文本框中输入了非数字型的字符，则当光标离开时，会弹出一对话框，告知用户输入错误。当单击 OK 按钮后，光标回到文本框，且文本框中的内容被高亮覆盖，这时即可重新输入数据。

(12) 用户可以按照第四章以及本章的内容，为程序加上减、乘、除的功能。这样，修改 myfirst 工程文件的工作就完成了。

第七章　菜单设计

广大用户对于菜单都是非常熟悉的。例如，我们现在正在使用的 Visual Basic 4.0就有 File、Edit 等八个菜单。菜单的出现给用户提供了极大的方便，不用使用按钮和选择框，只要下拉出一菜单就可以打开、关闭某一文件、对程序进行编辑等等。在 Visual Basic 中还有一弹出式菜单，当用户在 Form 窗口中或某一控件中单击一下鼠标右钮时会弹出一菜单，可以对控件进行剪切、删除等操作，如图7.1所示。


Microsoft Visual Basic - Project1 [design]
File
Edit
View
Insert
Run
Tools
Add-Ins
Help
1680,720
1455 x 495
Form1
Text1
Cut
Copy
Paste
Clear
Align to Grid
Properties
View Code
Project1
View Form
View Code
Form1
Form1
Properties - Form1
My Briefcase
Start
Project1
Microsoft Visu...
Properties - F...
Form1
9:52 PM

Text1 TextBox

Alignment	0 - Left Justify
BackColor	&H80000005&
BorderStyle	1 - Fixed Single
DataField	
DataSource	
DragIcon	(none)
DragMode	0 - Manual
Enabled	True
Font	MS Sans S

图7.1　Form 窗口中弹出式菜单

弹出式菜单在平时是隐含的，当要使用它时，只要单击一下鼠标右钮，即可弹出。

7.1　Menu Editor(菜单编辑器)窗口简介

用户在建立应用程序时，有时在一个 Form 窗口内为了方便，需要建立一菜单，可以选择一些命令来执行。这样不仅方便直观，而且还可以使用户设计的 Form 窗口更加标准。

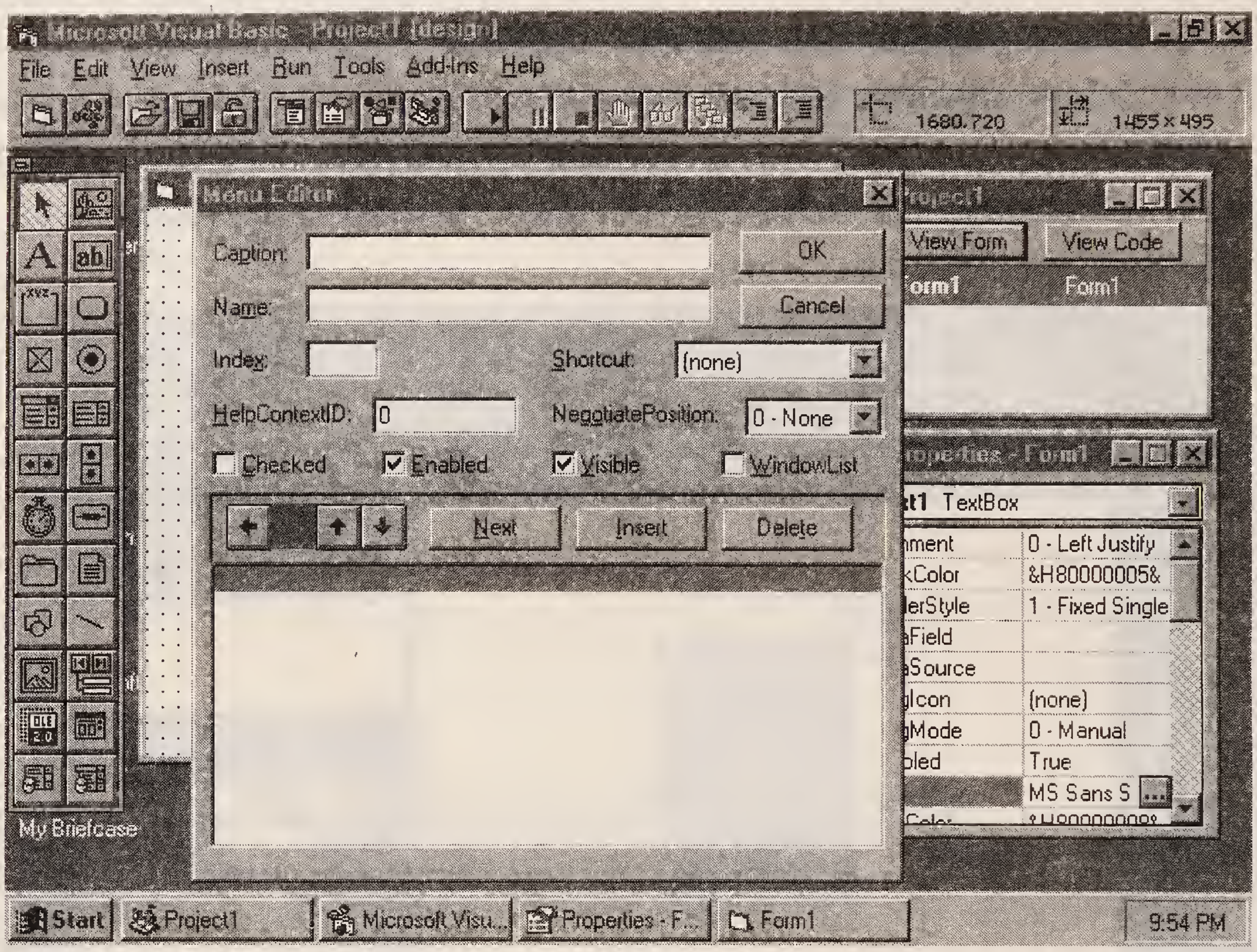
设计菜单是在菜单编辑器(Menu Editor)窗口内进行，要进入 Menu Editor 窗口，有两种途径。

(1) 先选择 File 菜单中的 New Project 命令，建立一个新工程文件。或者打开一个想要建立菜单的 Form 所在的工程文件，选择 Form，然后选择 Tools 菜单中的 Menu Editor 命令。

(2) 单击工具栏中的 Menu Editor 按钮。

要想关闭 Menu Editor，选择 OK 按钮接收设置即可。如想取消设置好的内容，单击 Cancel 按钮。

打开后的 Menu Editor 窗口如图7.2所示，分上下两部分。上半部分是用来设置属性，而下半部分则用来显示用户设置的菜单和菜单项。


Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
1680,720
1455 x 495
Menu Editor
OK
Cancel
Next
Insert
Delete
Project1
View Form
View Code
orm1
Form1
Properties - Form1
My Briefcase
Start
Project1
Microsoft Visu...
Properties - F...
Form1
9:54 PM

Caption:
Name:
Index:
Shortcut: (none)
HelpContextID: 0
NegotiatePosition: 0 - None
☐ Checked ☑ Enabled ☑ Visible ☐ WindowList

t1 TextBox

nment	0 - Left Justify
kColor	&H80000005&
lerStyle	1 - Fixed Single
aField	
aSource	
gIcon	(none)
gMode	0 - Manual
bled	True
	MS Sans S

图7.2 Menu Editor 窗口

下面我们结合一个设计好的 File 菜单介绍一下 Menu Editor 中各对话框及按钮等的作用。

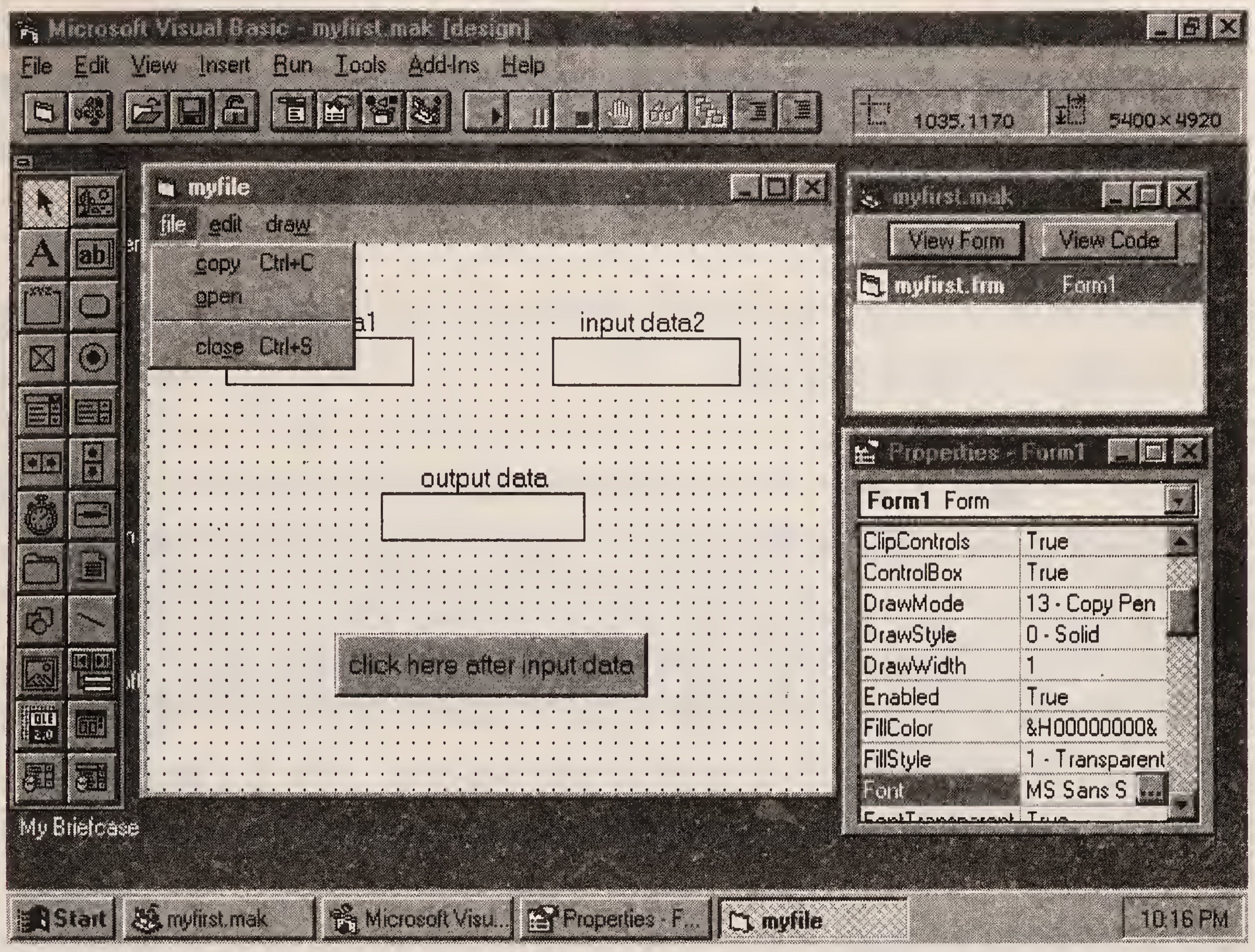
(1) Caption(标题)对话框

Caption 对话框，是一个文本框，用来输入用户建立的菜单的菜单名以及菜单中每个菜单项的名称。当在这个文本框中输入标题后，在用户所建立的菜单中，菜单名及各菜单项的名字将显示出来。

在图7.3设计好的 File 菜单中，会看到在 Open 命令与 Close 命令之间有一条线把它们分割开来，这条线叫分割条(Seperator bar)。要建立分割条，在 Caption 框体中只要输入连接符(—)即减号即可。

在图7.3中，我们还看到，在 Copy 命令中 C 字符下面有一横线(_)，这一横线表示 C 为 Copy 命令的访问(Access)键，即不必下拉菜单，只要在键盘上按 Alt+C 键(即 Alt 与 C 键同时按下) 即可。当用户在 Caption 框中输入菜单项或菜单名时即可建立访问键。具体做法是在访问键(如 C)前加一与号(&)即可。访问键可以是该菜单项的第一个字符，也可以是菜单项中间或最后的一个字符，在这里需要注意的是，不要为菜单名或任何菜单项建立相同的访问键，否则，只有第一次建立的那个访问键才是有效的。

如果用户在菜单中需要一个与号(&)，则需在 Caption 中键入两个 & 才可以，否则，& 后的那个字符会被设置为访问键。


Microsoft Visual Basic - myfirst.mak [design]
File Edit View Insert Run Tools Add-Ins Help
1035,1170
5400×4920
myfile
file edit draw
copy Ctrl+C
open
close Ctrl+S
input data2
output data
click here after input data
myfirst.mak
View Form
View Code
myfirst.frm Form1
Properties - Form1
Form1 Form
My Briefcase
Start
myfirst.mak
Microsoft Visu...
Properties - F...
myfile
10:16 PM

ClipControls	True
ControlBox	True
DrawMode	13 - Copy Pen
DrawStyle	0 - Solid
DrawWidth	1
Enabled	True
FillColor	&H00000000&
FillStyle	1 - Transparent
Font	MS Sans S

图7.3 File 菜单

(2) Name(控制名)对话框

Name 对话框也是一个文本框，用户可以用来输入各菜单名及菜单项的控制名。控制名是用来在进行代码编辑时标明身份的标志。在这里每一个菜单名及菜单项也都是控件。菜单设置中的每一项，不管是菜单名还是菜单项，甚至像分割条这样不参加代码编辑的分割符，只要在 Caption 框中输入过标题，都要为其取一个控制名。控制名并不在菜单中出现。

(3) Index(下标)对话框

Index 框为一文本框，用来为用户建立的控件数组设立下标，控件数组是一组享有同一控制名，而各自拥有不同的属性的控件。

(4) Short Cut(快捷键)对话框

Short Cut 对话框为一列表框，在其右侧有一向下的箭头，单击这一箭头，即会出现一列表框，列出了用户可以选择的各种快捷键。

(5) Help Context ID(帮助上下文 ID)对话框

Help Context ID 是一文本框，用户可以通过键入数字，来选择帮助文件中特定的页数。选择与该菜单上下文相关的帮助文本，可以通过 Help File 属性判定 Help 文件。

(6) Negotiate Position(可变位置)对话框

Negotiate Position 为一列表框，单击其右侧向下的箭头，即拉下一列表框。用户可以通过这一对话框来确定菜单在窗体中是否出现或怎样出现。

(7) Checked Box(复选框)

在图7.3中可以看到，在命令 cls 的左边出现一个"√"标记，这一标记为一选中标记。这一对话框允许用户设置某一菜单项是否可选，即表明菜单是否可以访问这一菜单项。

(8) Enabled 框

Enabled 框，是一个用来设置该菜单项是否为可执行的框体。即用户是否想让这一菜单项响应某一事件，即如果 Enabled 设置为 False，则菜单不能访问这一菜单项，也就是这一菜单命令是无效的。如果某一菜单名的 Enabled 属性为 False，则整个菜单命令都是无效的。如一菜单项是无效的，可以删除这一菜单项，或把它设置为灰色的。

(9) Visible（可见）框体

如果 Visible 为 False，则选中的菜单项在菜单中是不可见的；如果某一菜单名的 Visible 属性为 False，则整个菜单都是不可视的。

(10) Window List 框体

在 Window List 框体中，用户可以决定是否在使用多文档应用程序时，使菜单控件中有一包含打开的多文档文件子窗口的列表框。

(11) 箭头按钮

在图7.2中，我们看到 Menu Editor 窗口中间有四个箭头按钮。

上下箭头按钮用于将选中的菜单项向上或向下移动一位，改变一下菜单中各菜单项的顺序。也就是说，首先选中一菜单项，这一菜单项被高亮显示。当单击上箭头按钮时，该菜单项与上一菜单项互换一下位置。单击向下的箭头时，则该菜单项与下一菜单项互换一下位置。

左右箭头用于将菜单项向左或向右缩进四格距离。特别是右键头，当建立菜单名以后，再键入菜单项时，要单击向右的箭头，使其向右移动，缩进四格距离，表示其为菜单中的菜单项。若某一菜单项为一子菜单，它下面又有若干菜单项，则在建立子菜单名后，再键入菜单项时，还要单击右键头按钮，使其向右缩进四格距离。关于这些内容，在后面建立一个菜单例子中，再详细为读者演示。

在 Menu Editor 窗口的中间部分还有三个按钮，分别是 Next、Insert 和 Delete。

Next 按钮表示移到下一个选择项。即当用户把一个菜单名或菜单项的属性设置完以后。单击此按钮，即可设置下一个菜单名或菜单项。

Insert 按钮用于插入菜单项。即当用户建立多个菜单项后，想在某一菜单项前添加一新的菜单项时，首先单击选中的菜单项，当这一菜单项被高亮显示后，单击 Insert 按钮。这时选中的菜单项将向下移动一行，上面空出一行，用户就可以在这一行设置新的菜单项了。

Delete 按钮用于删除某一菜单名或菜单项。首先，单击选中的菜单项或菜单名，然后单击 Delete 按钮，即可删除。

在 Menu Editor 窗口的下半部分是一空白框体，在这一框体中将列出用户为某一窗体所设计的所有菜单名和菜单项。

用户在设计菜单时，在 Menu Editor 窗口上每设置完一个菜单名或菜单项，这一菜单名或菜单项就马上在下面的大框体中显示出来。

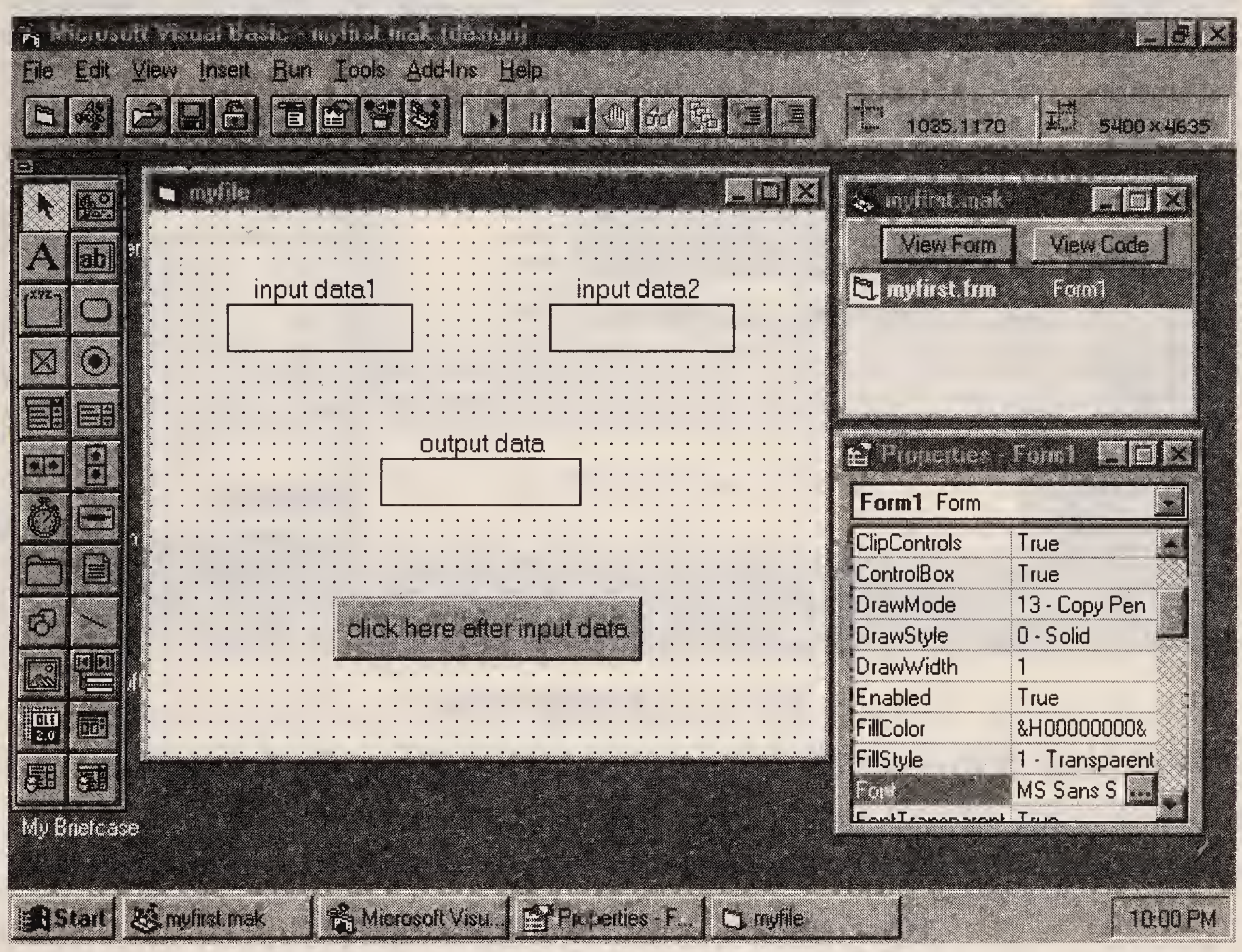
7.2 建立菜单

要想建立一个菜单，用户得首先列出菜单名及菜单项，在 Menu Editor 窗口中对菜单名及各菜单项进行编辑，设计属性。设计完以后，再把菜单中各控件（即各菜单名与菜单项）与代码连接起来。

在本书的第四章，我们建立了一个简单的应用程序，在这里就以它为例，为其 Form 窗口建立一菜单，我们在这里只是给大家介绍怎样建立一个菜单，菜单里的大多数命令我们就不再为其编写代码了，只是为几个常用命令设计代码，把代码与命令连接起来。

7.2.1 建立菜单条

首先，打开 myfirst.mak 工程文件。选择 File 菜单中的 Open Project 命令，即出现一对话框，在对话框中键入 C:\VB\myfirst.MAK 命令，再回车，或单击 OK 按钮，则打开了 myfirst.MAK 工程文件。这时 Form 窗口并未打开，单击 Project 窗口中的 View Form 命令，屏幕上即出现 Form 窗口，这时 Form 窗口的标题为 myfile，如图7.4所示。


Microsoft Visual Basic - myfirst.mak [design]
File Edit View Insert Run Tools Add-Ins Help
1035,1170
5400 x 4635
myfile
input data1
input data2
output data
click here after input data
myfirst.mak
View Form
View Code
myfirst.frm
Form1
Properties - Form1
My Briefcase
Start
myfirst.mak
Microsoft Visu...
Properties - F...
myfile
10:00 PM

Form1 Form

ClipControls	True
ControlBox	True
DrawMode	13 - Copy Pen
DrawStyle	0 - Solid
DrawWidth	1
Enabled	True
FillColor	&H00000000&
FillStyle	1 - Transparent
Font	MS Sans S

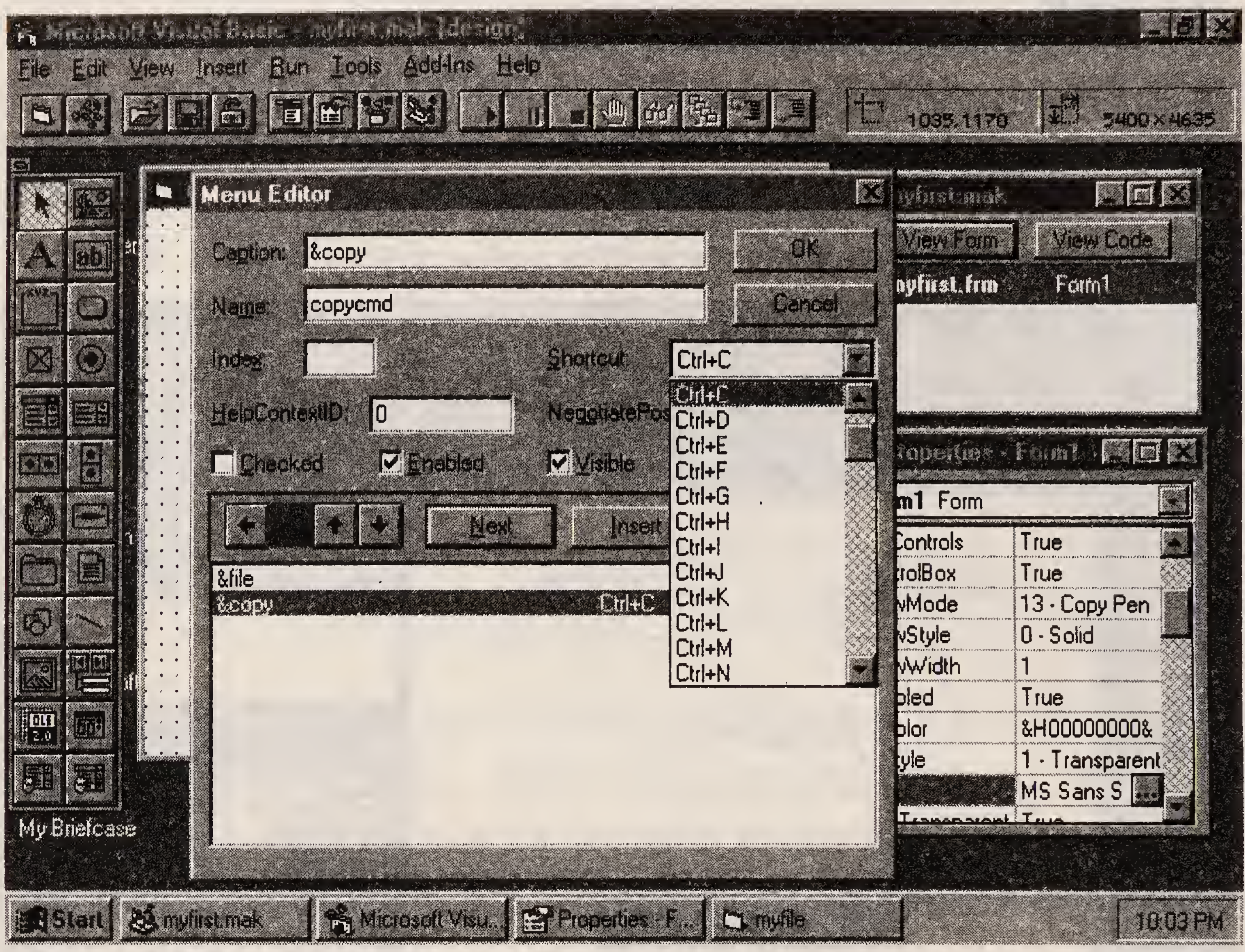
图7.4 myfirst 工程文件

这时就可以为 myfirst 窗体窗口设计菜单了。

选择 Tools 菜单中的 Menu Editor 命令，或者单击工具栏中的 Menu Editor 按钮。在屏幕的中间即出现 Menu Editor 窗口，如图7.2所示。

在 Caption 框体中，我们会看到一活动光标，在这里键入 File 菜单名。注意，我们想把 File 中的 F 作为访问键，这时要在 Caption 中键入 &File。这样在 File 菜单条中会在 F 字符下出现下横线，用户可以按 Alt+F 键来访问 File 菜单。然后，单击 Tab 键，活动光标即移到 Name 框体中，键入 mnuFile，表示 File 菜单名的控件名为 mnuFile(File 菜单)。单击 Next 按钮，则在 Menu Editor 窗口下的大框体中马上出现菜单名 File。同时，Caption 与 Name 框体又变为空白，并且光标又回到 Caption 框体中。

在 Caption 框体中输入 &Copy，单击 Tab 键，在 Name 框体中输入 Copycmd，表示 Copy 为一菜单项。在 Shortcut 框中单击下箭头，从列出的各快捷键中选择 Ctrl+C 为快捷键，如图7.5所示。然后单击一下命令按钮中的向右箭头（在 Menu Editor 窗口中部。在这里，它是一黑色框体，没有标明箭头。在向左箭头和向上箭头中间的那个黑色框体就是）。


Caption: &copy
Name: copycmd
Index:
Shortcut: Ctrl+C
HelpContextID: 0
☐ Checked ☑ Enabled ☑ Visible

&file
&copy Ctrl+C

Microsoft Visual Basic - myfirst.mak [design]
File Edit View Insert Run Tools Add-Ins Help
1035,1170
5400 x 4635
Menu Editor
OK
Cancel
NegotiatePos
Next
Insert
Ctrl+C
Ctrl+D
Ctrl+E
Ctrl+F
Ctrl+G
Ctrl+H
Ctrl+I
Ctrl+J
Ctrl+K
Ctrl+L
Ctrl+M
Ctrl+N
myfirst.mak
View Form
View Code
myfirst.frm
Form1
Properties - Form1
Form1 Form
Controls True
ControlBox True
DrawMode 13 - Copy Pen
DrawStyle 0 - Solid
DrawWidth 1
Enabled True
FillColor &H00000000&
FillStyle 1 - Transparent
MS Sans S
My Briefcase
Start
myfirst.mak
Microsoft Visu...
Properties - F...
myfile
10:03 PM

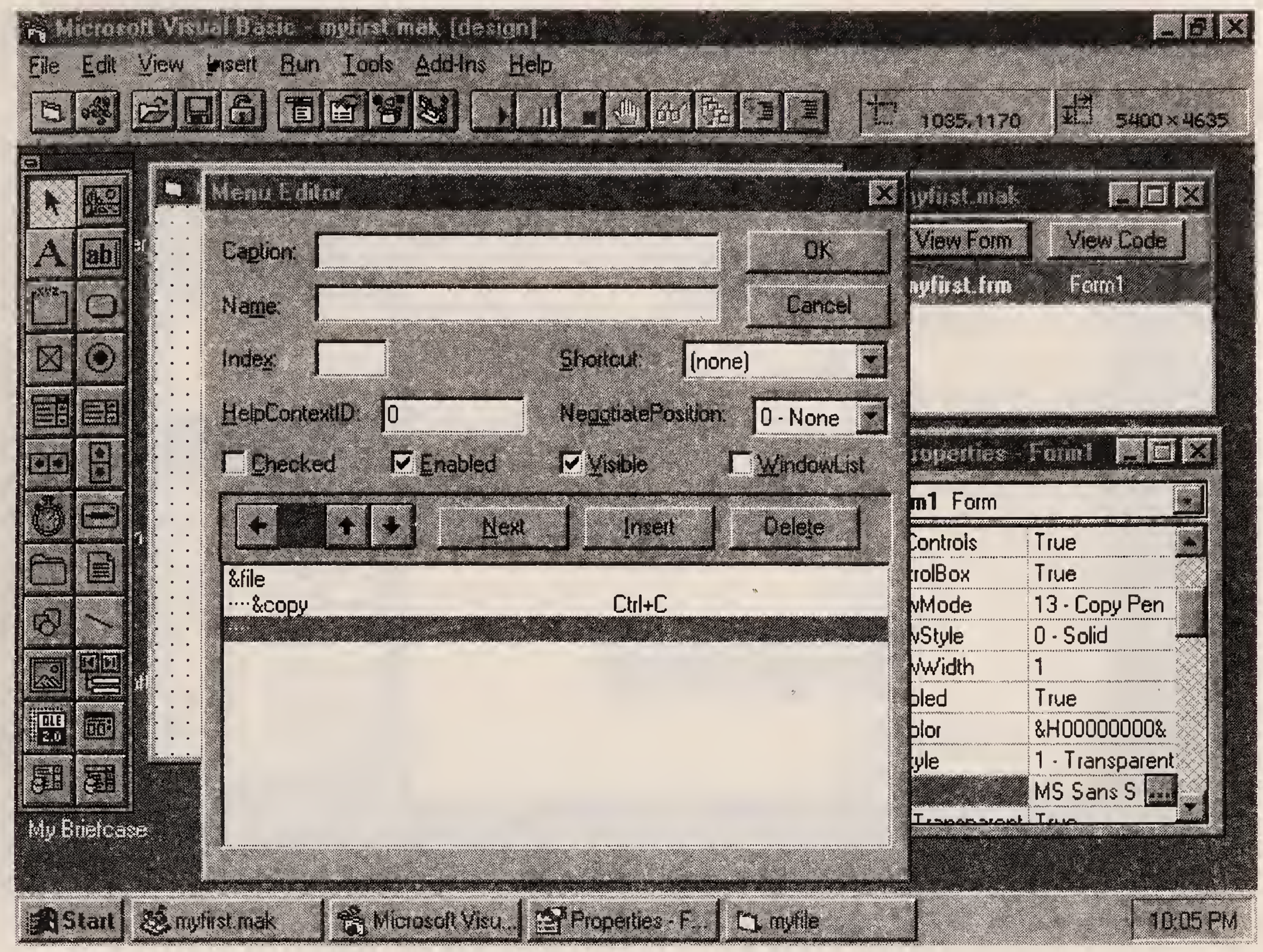
图7.5 选择快捷键对话框

然后单击 Next 按钮，Copy 菜单项马上在大框体中显示出来。由于 Copy 为 File 菜单中的一菜单项，所以我们刚才单击右箭头，使其向右缩进四格。如图7.6所示。

读者可以按照以上步骤把下列各菜单项输入到 Menu Editor 中。

表7.1 输入到 Menu Editor 窗口中的菜单项

菜单项	快捷项	控制名	下标
& File		MnuFile	
....&Copy	Ctrl+C	Copycmd	
....&Open		Opencmd	
...._		Linecmd	
....Clo&se	Ctrl+S	Closecmd	
&Edit		MnuEdit	
....&1Font	F1	Fontcmd	8
....&2Font	F2	Fontcmd	12
....&3Font	F3	Fontcmd	14
Dra&W		MnuDraw	
....Colo&r		SubColor	
........&Red	Ctrl+R	Redcmd	
........&Blue	Ctrl+B	Bluecmd	


Microsoft Visual Basic - myfirst.mak [design]
File Edit View Insert Run Tools Add-Ins Help
1035, 1170
5400 x 4635
Menu Editor
OK
Cancel
Next
Insert
Delete
myfirst.mak
View Form
View Code
myfirst.frm
Form1
Properties - Form1
My Briefcase
Start
myfirst.mak
Microsoft Visu...
Properties - F...
myfile
10:05 PM

Caption:
Name:
Index:
Shortcut: (none)
HelpContextID: 0
NegotiatePosition: 0 - None
☐ Checked ☑ Enabled ☑ Visible ☐ WindowList

```
&file
····&copy                     Ctrl+C
····
```

m1 Form

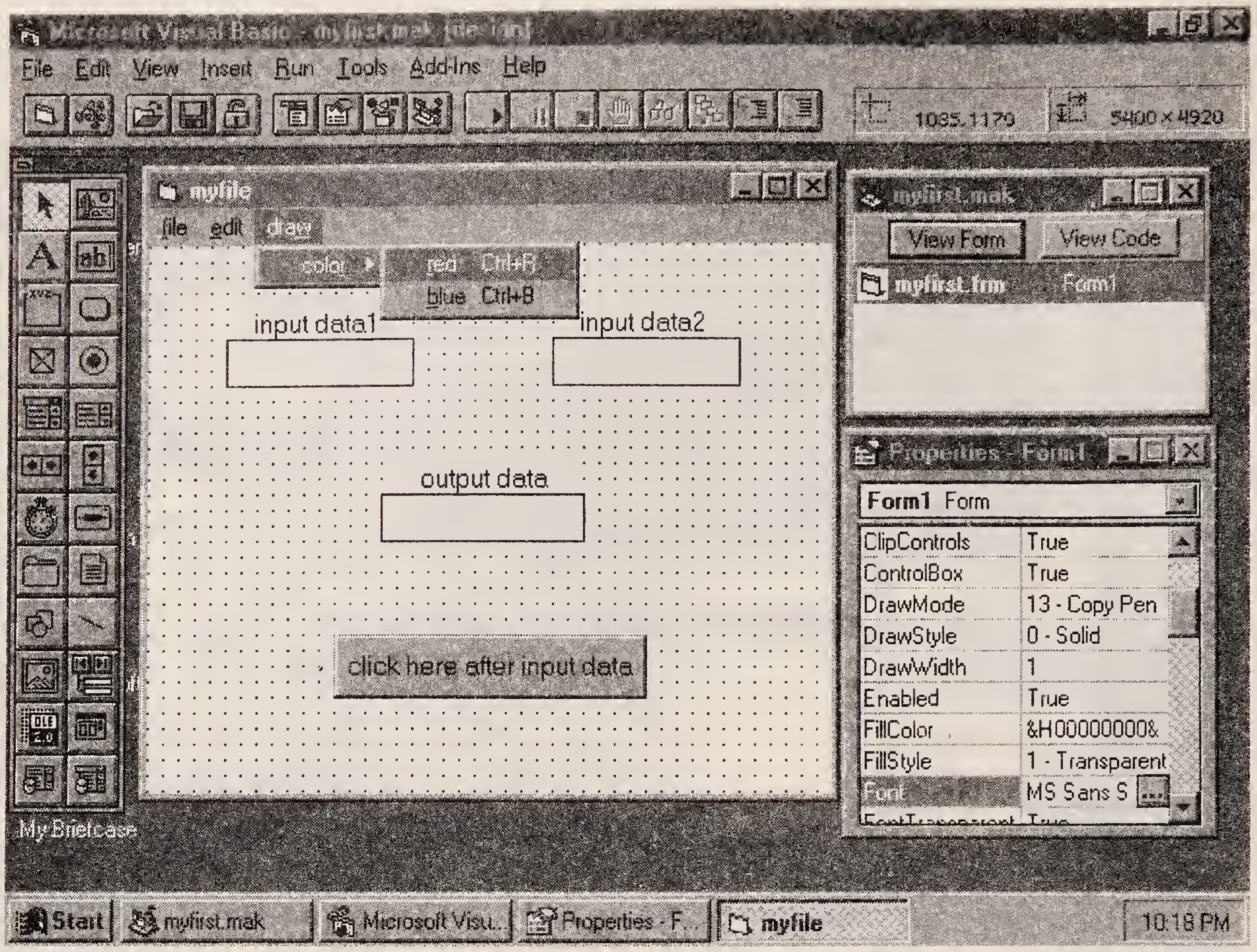
ontrols	True
rolBox	True
wMode	13 - Copy Pen
wStyle	0 - Solid
wWidth	1
bled	True
olor	&H00000000&
yle	1 - Transparent
	MS Sans S

图7.6　File 菜单名及 Copy 菜单项

读者在将表7.1中的内容往 Menu Editor 中输入时，注意到在 Edit 菜单中的各菜单项 1Font、2Font 和3Font 具有相同的控件名 Fontcmd。这是因为它们是一控件数组，各菜单项享有同一控件名和同一过程。注意在输入1Font、2Font 及3Font 时，在 Caption 及 Name 框中键入标题及控制名后，要单击一下 Tab 键，使光标移到 Index 框体，键入下标。1Font、2Font 及3Font 的下标分别为8，12，14。同时，我们希望在1Font 菜单项的左边出现一标记“√”，这一属性并不改变菜单的效果，也不影响事件过程对任何对象的作用结果，它只是设置选项旁的选择符号，来表示当前该项所表现的程序功能是可以执行的。设置过程是先在 Menu Editor 下面的大框体中，选择1Font，然后单击 Checked 选项左边的小框体，当小框体中出现“√”时，就表示1Font 的 Checked 属性为 True。

在这些菜单项中。读者也会看到，即使是分割条(—)也是有控件名的，在本例中为 Linecmd。如果用户忘了给任何一个菜单项输入控件名，在关闭 Menu Editor 窗口时，Visual Basic 都会弹出一对话框，提醒用户某一菜单项没有设置菜单名。用户只要单击 OK 按钮，补上控件名即可。

在图7.7中，我们看到在 Draw 菜单中有一个 Color 子菜单。当在 Form 窗口中打开 Draw 菜单时，会发现 Color 菜单项右侧有一向右的箭头，当高亮移到 Color 菜单项时，会在其右侧弹出一子菜单。在设计子菜单时，如表7.1所示，子菜单中的菜单项要在子菜单名下往右缩进四格，即比主菜单名往右缩进八格。Visual Basic 允许用户设计的菜单中嵌套子菜单，但最多只许嵌套五层。一般如果没有必要的话，用户在设计菜单时最好不要设计多层子菜单。否则，会使整个菜单变的很复杂，操作起来也很不方便。


File
Edit
View
Insert
Run
Tools
Add-Ins
Help
1035,1170
5400 x 4920
myfile
file
edit
draw
color
red Ctrl+R
blue Ctrl+B
input data1
input data2
output data
click here after input data
myfirst.mak
View Form
View Code
myfirst.frm
Form1
Properties - Form1
My Briefcase
Start
myfirst.mak
Microsoft Visu...
Properties - F...
myfile
10:18 PM

Form1 Form

ClipControls	True
ControlBox	True
DrawMode	13 - Copy Pen
DrawStyle	0 - Solid
DrawWidth	1
Enabled	True
FillColor	&H00000000&
FillStyle	1 - Transparent
Font	MS Sans S

图7.7 Form 窗口中的 Color 子菜单

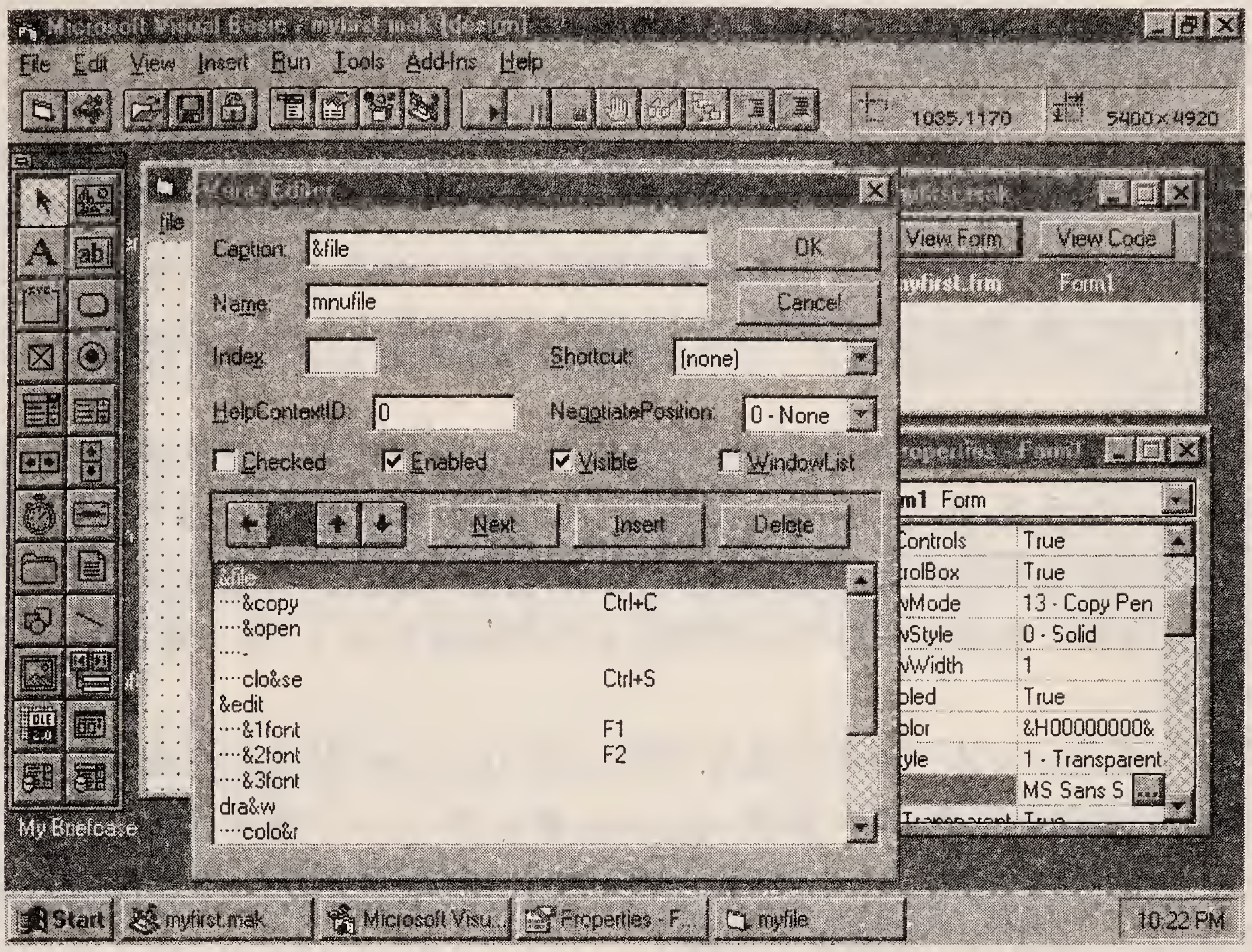
下面我们再设计一不可视(invisible)菜单。将鼠标移到列有菜单项的大文本框中所有菜单项的最下面，选择一空白行，单击鼠标。单击 Caption 框体，键入 &Do，单击 Tab 键，在 Name 框体中输入 mnuDo。然后将鼠标移到 Visible 框体。这时 Visible 框体中有一“√”号，表示 Visible 缺省值为 True。单击鼠标，“√”消失，表示这时 Visible 值为 False。如果这时 Do 为缩进格式，则单击命令按钮中的左箭头，使 Do 与左框对齐，表示它为菜单名，单击 Next 按钮。在 Caption 和 Name 框中分别输入 &Paste 与 Pastecmd，单击右按钮，再单击 Next 命令。再在 Caption 和 Name 框中分别输入 Lis&t 与 Listcmd 即可。这样，在 Form 窗口中，我们就看不到这个菜单了。

我们还要把 File 菜单中的 Copy 和 Open 命令设置为不可执行的。在 Menu Editor 下面的列表框中找到 Copy 菜单项，使其被高亮显示。这时 Caption 和 Name 框中将分别出现 &Copy 与 Copycmd。把鼠标指针移到 Enabled 框体上，我们发现这时 Enabled 框体内有一“√”号，表示其缺省值为 True，单击一下鼠标，这时“√”号消失了，表明其当前值为 False。再找到 Open 菜单项，重复以上命令。这样，在 Form 窗口中的 File 菜单中 Open 和 Copy 将被涂为灰色，并且单击鼠标时不做任何响应，即不执行这一命令。

这时，菜单条就设计好了，设计好菜单条的 Menu Editor 窗口如图7.8所示。

设计完各菜单项以后，就要关闭 Menu Editor 窗口，重新回到 Form 窗口。如果用户在设计菜单条时有什么错误，在关闭 Menu Editor 窗口时，Visual Basic 会弹出一对话框，告诉用户什么地方设置错误。用户只要单击对话框中的 OK 按钮，这一对话框就会消失。然后只要对设置错误的地方进行修改就可以了。

当各菜单项均准确无误地设置完以后，单击 Menu Editor 窗口中的 OK 按钮即可。如果用户不想运用这些已设置完属性的各菜单项，单击 Cancel(取消)按钮就可以了。


Microsoft Visual Basic - myfirst.mak [design]
File Edit View Insert Run Tools Add-Ins Help
1035,1170
5400 x 4920
Menu Editor
OK
Cancel
Next
Insert
Delete
myfirst.mak
View Form
View Code
myfirst.frm
Form1
Properties - Form1
My Briefcase
Start
myfirst.mak
Microsoft Visu...
Properties - F...
myfile
10:22 PM

Caption: &file

Name: mnufile

Index:

Shortcut: (none)

HelpContextID: 0

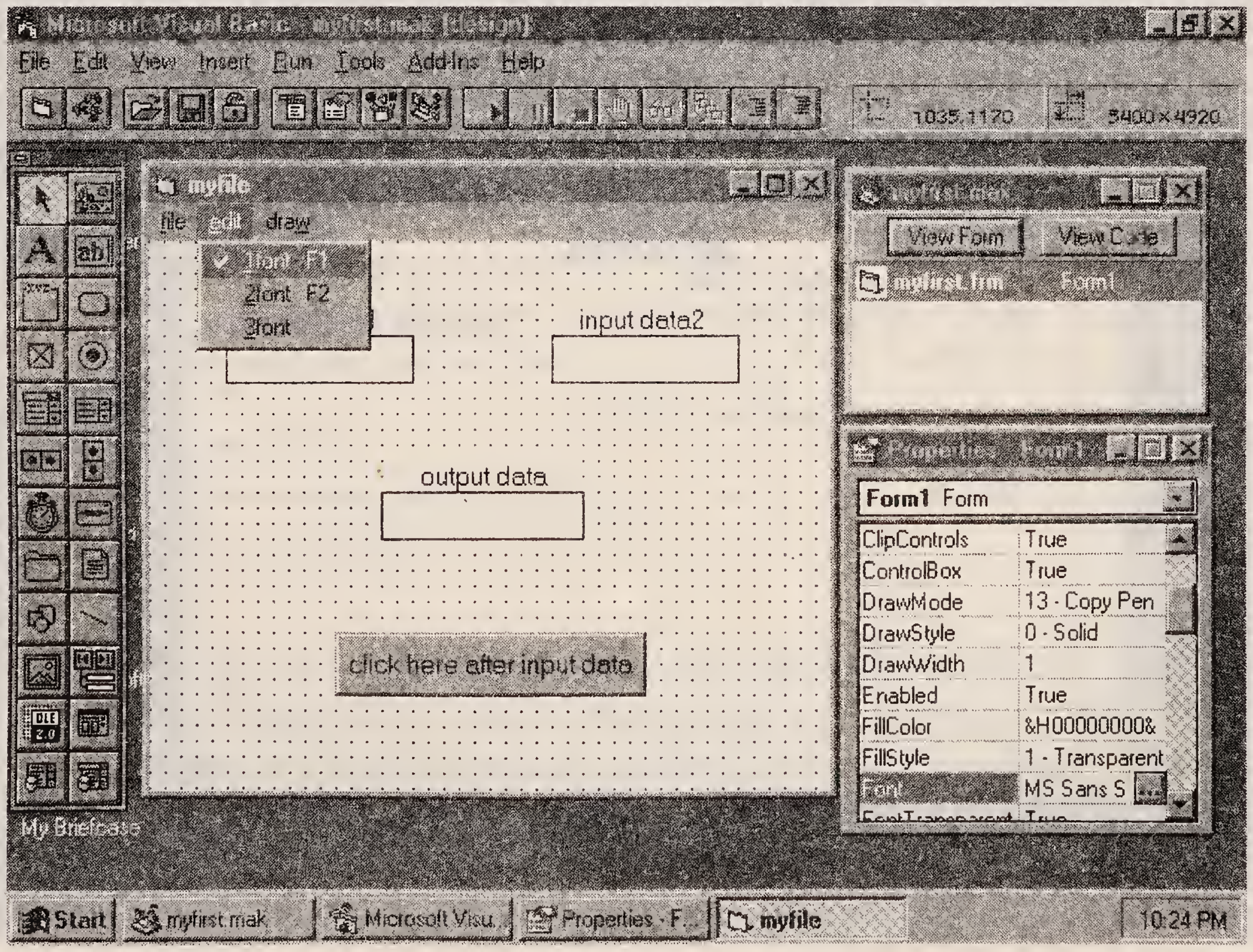
NegotiatePosition: 0 - None

☐ Checked　☑ Enabled　☑ Visible　☐ WindowList

```
&file
····&copy                Ctrl+C
····&open
····-
····clo&se               Ctrl+S
&edit
····&1font               F1
····&2font               F2
····&3font
dra&w
····colo&r
```

图7.8　设计好菜单条的 Menu Editor 窗口

当 Menu Editor 窗口消失以后，屏幕上又显示出了 Form 窗口，如图7.9所示。我们从图7.9中看到在 Form 窗口的上部出现一菜单条。由于 Do 菜单是隐含的，在窗口中看不到它。当把鼠标移到 Edit 菜单上单击一下，即下拉出一菜单。File 菜单以及 Draw 菜单分别在图7.3以及图7.7中。


Microsoft Visual Basic - myfirst.mak [design]
File Edit View Insert Run Tools Add-Ins Help
1035,1170
5400 x 4920
myfile
file
edit
draw
1font F1
2font F2
3font
input data2
output data
click here after input data
myfirst.mak
View Form
View Code
myfirst.frm
Form1
Properties - Form1
Form1 Form
My Briefcase
Start
myfirst.mak
Microsoft Visu...
Properties - F...
myfile
10:24 PM

ClipControls	True
ControlBox	True
DrawMode	13 - Copy Pen
DrawStyle	0 - Solid
DrawWidth	1
Enabled	True
FillColor	&H00000000&
FillStyle	1 - Transparent
Font	MS Sans S

图7.9　Form 窗口上的菜单条以及 Edit 菜单

7.2.2　把代码连接到菜单项上

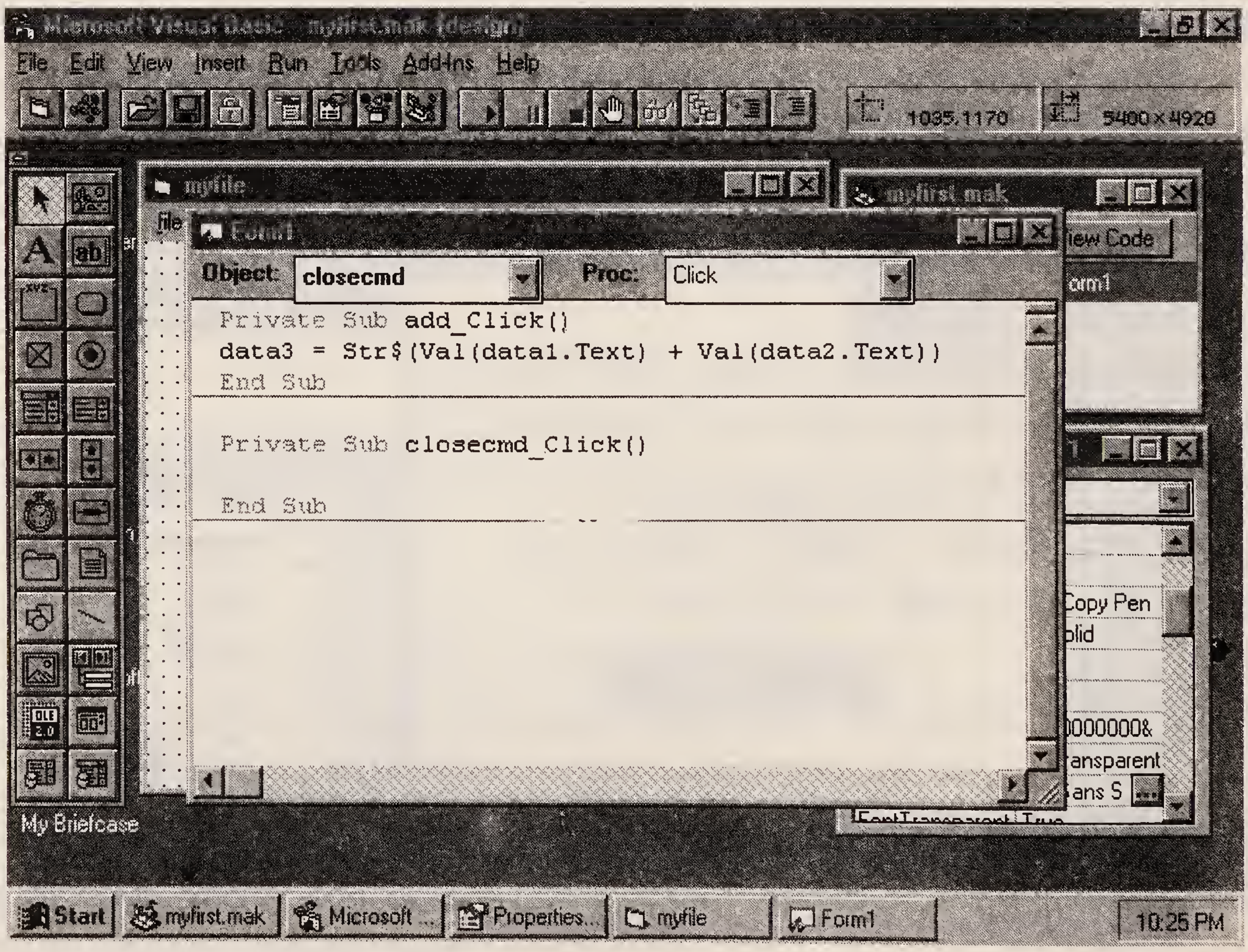
在 Visual Basic 中，把每一菜单项，甚至包括分割符都看作一控件。每一菜单项都要响应某一事件过程。一般来说，菜单命令都响应鼠标单击(Click)事件，即每个菜单项都拥有一个事件处理例程 Name _ Click()。每当单击菜单项时，Visual Basic 就调用 Name _ Click()，执行这一处理器中的代码，即命令。下面我们所要做的工作就是为 Close、Color 以及 Font 控件数组命令编写代码。

编写代码是在 Code 窗口内进行的。首先在 Form 窗口中单击菜单条，在下拉菜单中选择要连接代码的菜单项，然后单击这一菜单项，在屏幕中即出现 Code 窗口，并在窗口中出现这一菜单项的控件名与相应事件组成的事件处理例程的过程头与过程尾。用户只要在过程头与过程尾之间输入想要这一事件完成的任务的代码就可以了。这一菜单项与代码的连接过程就完成了。

如果用户想同时为另一菜单项编写代码，这时在 Object 列表框中选择菜单项控件名，在 Procedure 列表框中选择 Click 事件，在 Code 窗口中，上一个菜单项过程代码的下面，即出现这一新的菜单项的过程头与过程尾，在其中嵌入过程代码即可。如果有多个菜单项需要与代码过程连接 ，就得多次重复以上步骤。

(1) 为 Close 菜单项编写代码过程

Close 菜单项在 File 菜单中，因此，首先单击 File 菜单条，在下拉菜单中单击 Close，这时，屏幕中间出现 Code 窗口，且同时出现 Closecmd _ Click 事件的过程头与过程尾，如图7.10所示。


Microsoft Visual Basic - myfirst.mak [design]
File
Edit
View
Insert
Run
Tools
Add-Ins
Help
1035,1170
5400 x 4920
myfile
myfirst.mak
View Code
Form1
Copy Pen
Solid
00000008
Transparent
Sans S
My Briefcase
Start
myfirst.mak
Microsoft ...
Properties...
myfile
Form1
10:25 PM

Object: closecmd

Proc: Click

```
Private Sub add_Click()
data3 = Str$(Val(data1.Text) + Val(data2.Text))
End Sub

Private Sub closecmd_Click()

End Sub
```

图7.10　Closecmd _ Click 事件的过程头与过程尾

我们希望在执行 Close 这一命令时，将结束程序的运行，重新回到 Design（设计）模式下。而结束程序运行的命令是 End。所以我们把光标移到 Closecmd _ Click 事件的过程头与过程尾之间，键入 End 命令。即这一过程代码为：

```
Private Sub Closecmd _ Click()
    End
End Sub
```

这样，当运行程序时，如果用户想结束程序的运行，只要选择 File 菜单中的 Close 命令就可以了。

(2) 为 Color 子菜单中的 Red 和 Blue 命令添加代码

首先单击 Form 窗口中的 Draw 菜单，将光标移到 Color 菜单条上，即在其右侧出现 Red 和 Blue 菜单命令，单击 Red 命令，在出现的 Code 窗口中的 Redcmd _ Click 事件过程头与过程尾中添加代码：

```
Private Sub Redcmd _ Click()
   Form1. BackColor=QBColor(4)
End Sub
```

同理，在 Blue 命令的代码过程中输入下列代码

```
Private Sub Bluecmd _ Click()
   Form1. BackColor=QBColor(1)
End Sub
```

这样，当用户在运行程序时单击 Color 子菜单中的各菜单命令，即可将 Form1的背景色变为红色或蓝色。

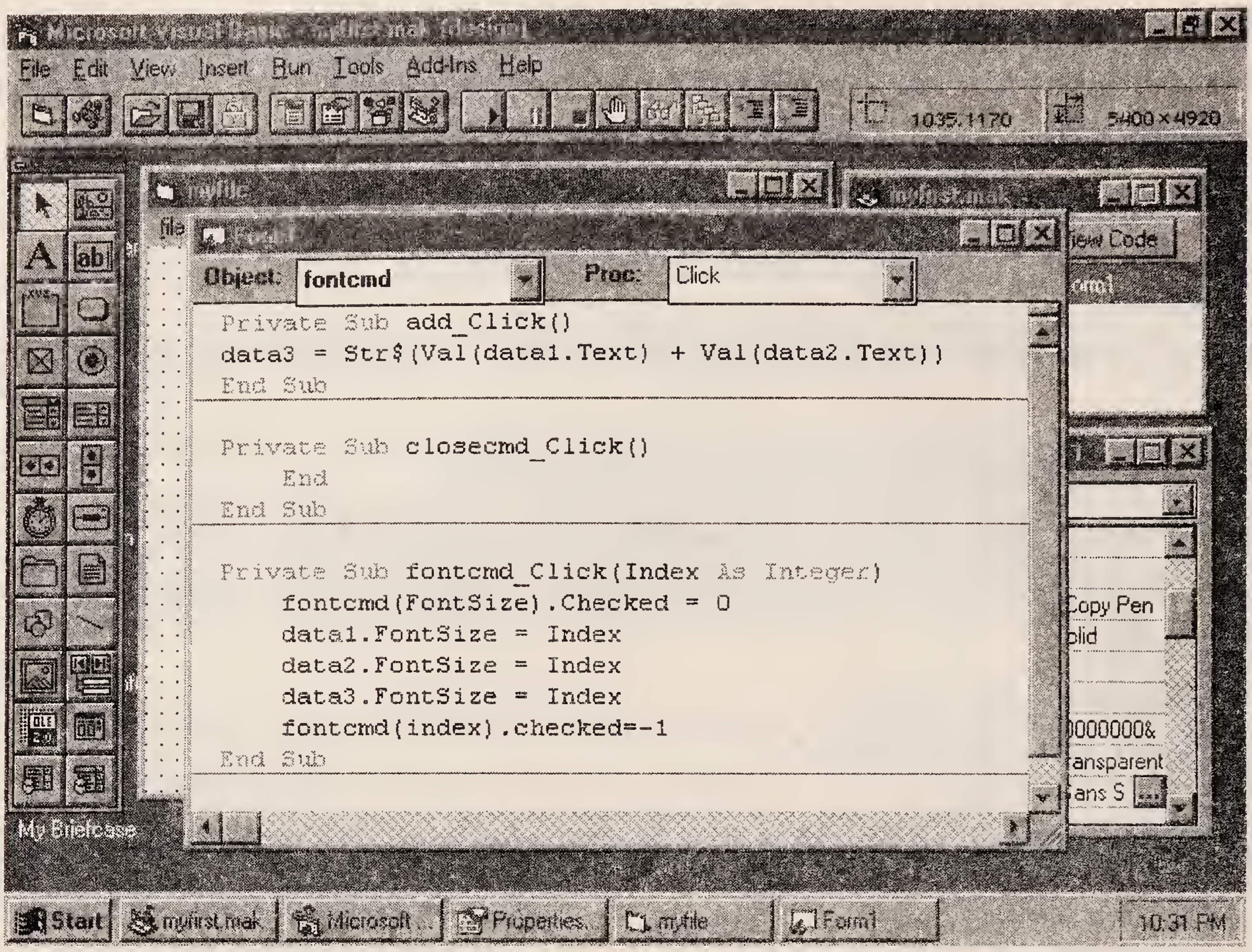
(3) 为1Font、2Font、3Font 控制数组编写代码

1Font、2Font、3Font 都在 Edit 菜单中，它们拥有相同的控件名，且分享同一代码过程。所以只要编一段代码就可以了。我们希望在执行这些命令时，可以改变文本框中输入输出字体的大小。在 Code 窗口中的 Object 列表框中选择 Fontcmd 控件名，在 Procedure 列表框中选择 Click 事件。在 Fontcmd _ Click 事件的过程头与过程尾中嵌入代码过程。

```
Private Sub Fontcmd _ Click()
   fontcmd(FontSize). Checked=0
   data1. FontSize=Index
   data2. FontSize=Index
   data3. FontSize=Index
   fontcmd(Index). Checked=-1
End Sub
```

这段代码的意思是，对于当前的字体大小不进行检查，然后改变字体大小，对改变后的字体进行检查。当程序开始运行时，字体大小的当前值为12，这在设计工程文件 myfirst 时就已设定了。

这样，File 菜单中的 Close 命令，Edit 菜单中的 Font 命令以及 Color 子菜单中各命令的代码就编辑完了。编辑完代码的 Code 窗口，如图7. 11所示。


Microsoft Visual Basic - myfirst.mak [design]
File Edit View Insert Run Tools Add-Ins Help
1035,1170
5400 x 4920
myfile
Form1
myfirst.mak
View Code
Copy Pen
My Briefcase
Start
myfirst.mak
Microsoft...
Properties
myfile
Form1
10:31 PM

Object: fontcmd Proc: Click

```
Private Sub add_Click()
data3 = Str$(Val(data1.Text) + Val(data2.Text))
End Sub

Private Sub closecmd_Click()
    End
End Sub

Private Sub fontcmd_Click(Index As Integer)
    fontcmd(FontSize).Checked = 0
    data1.FontSize = Index
    data2.FontSize = Index
    data3.FontSize = Index
    fontcmd(index).checked=-1
End Sub
```

图7.11 编辑完代码的 Code 窗口

代码编辑完成以后，就可以将其关闭，回到 Form 窗口。在这里，用户要注意，菜单中的各命令在程序运行时打开的 Form 运行窗口中才是有效的。在运行前的 Form 窗口中单击某一菜单命令时，只是打开 Code 窗口。

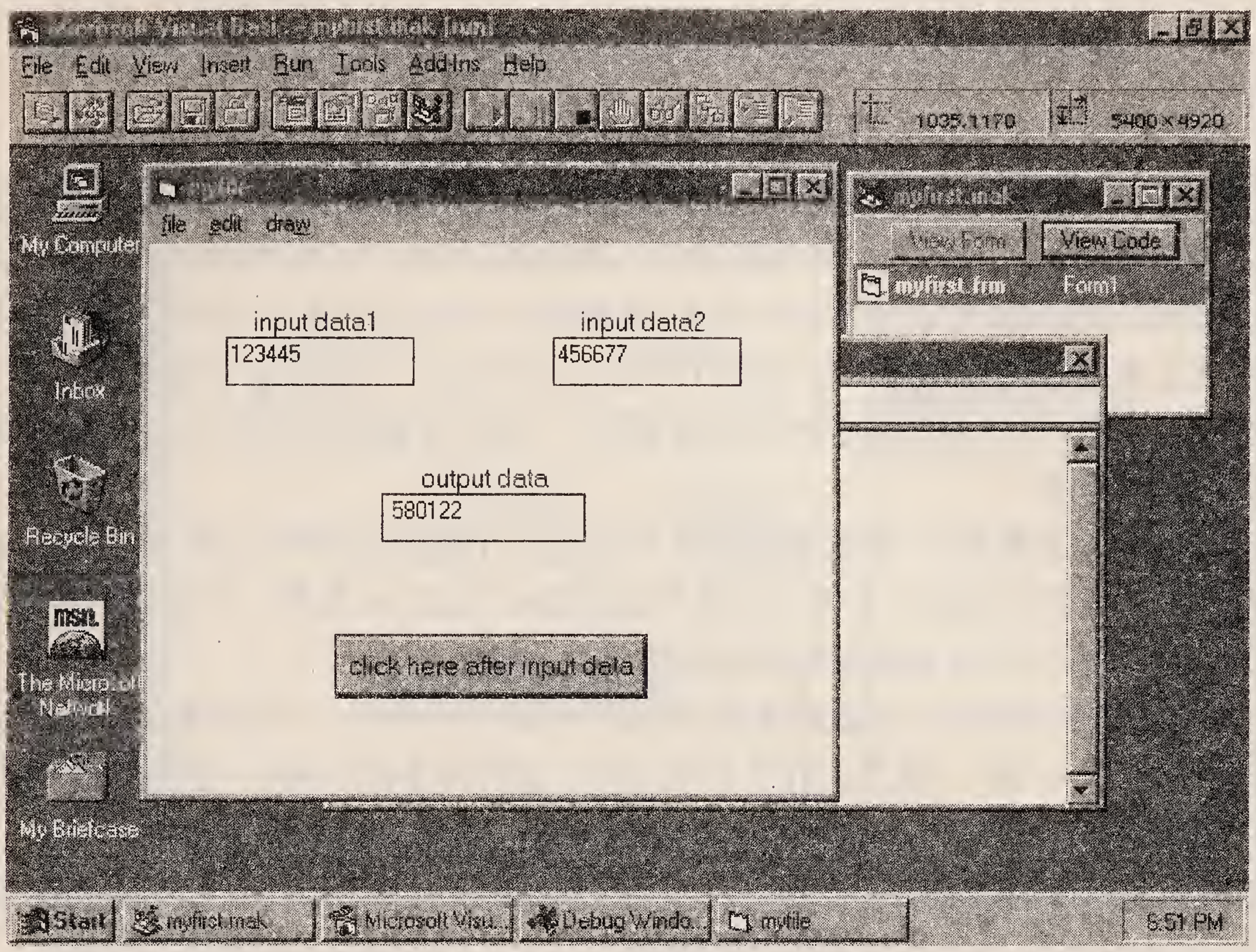
如果菜单中的各命令执行有误，用户可以重新打开 Code 窗口对代码过程进行修改。这样，一个菜单就设计完成了。

7.3 试验菜单命令

当设计完一个菜单后，我们就要运行一个程序，然后检查编写后代码的各菜单命令是否有效。

首先，单击 Visual Basic 菜单条中 Run 菜单中的 Start 命令，或单击工具栏中的 Start 按钮，运行 myfirst 工程文件。

在 Form 窗口中的 data1及 data2文本框中输入数据，单击命令按钮，在 data3文本框中输出两数之和。选择 Edit 菜单中的1Font 命令，2Font 命令以及3Font 命令，就会发现 data1、data2以及 data3文本框中的字符随着不同的命令而改变字体大小。并且随着字体的扩大，文本框的上下高度也相应变大，但框体长度并未改变。当我们把字体又变小时，文本框并未缩小。如图7.12、图7.13所示。当然用户也可以通过选择其它属性来改变字体的形式。

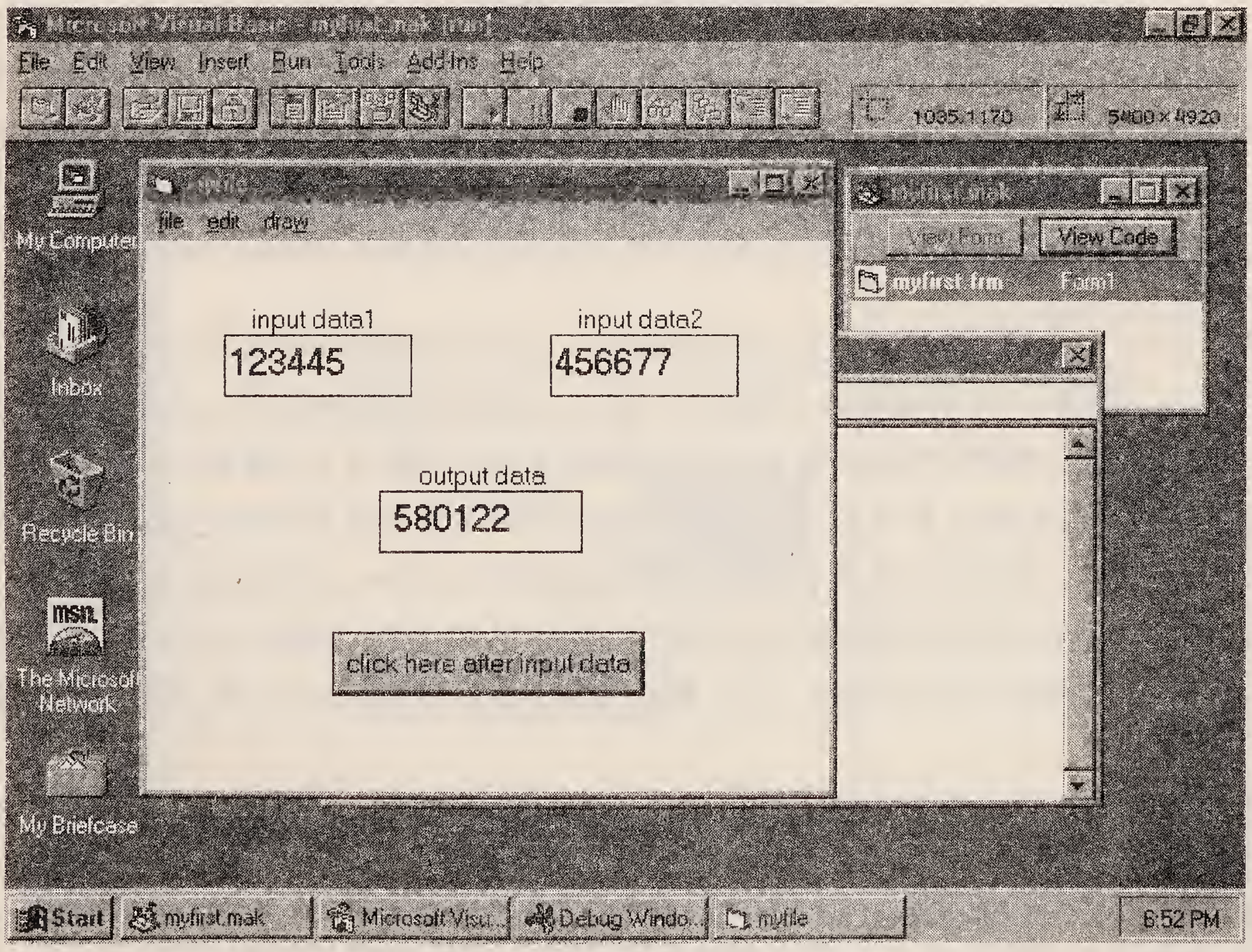

Microsoft Visual Basic - myfirst.mak [run]
File Edit View Insert Run Tools Add-Ins Help
1035,1170
5400 x 4920
My Computer
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
myfile
file edit draw
myfirst.mak
View Form
View Code
myfirst.frm
Form1
click here after input data
Start
myfirst.mak
Microsoft Visu...
Debug Windo...
myfile
5:51 PM

input data1: 123445

input data2: 456677

output data: 580122

图7.12　选择1Font 命令时文本框中字体变小


Microsoft Visual Basic - myfirst.mak [run]
File Edit View Insert Run Tools Add-Ins Help
1035,1170
5400 x 4920
My Computer
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
myfile
file edit draw
myfirst.mak
View Form
View Code
myfirst.frm
Form1
click here after input data
Start
myfirst.mak
Microsoft Visu...
Debug Windo...
myfile
6:52 PM

input data1: 123445

input data2: 456677

output data: 580122

图7.13　选择3Font 命令时文本框中字体变大

如果用户想结束程序的运行，只要选择 File 菜单的 Close 命令就可以了。

7.4 上托菜单

正如我们在开始所讲的那样，在 Visual Basic 中允许出现快速菜单，这点对用户来说，真是提供了很大的方便。一般来说，菜单都在窗口的顶部。有时，用户需要执行菜单中的某一命令，就得把鼠标指针移到窗口顶部打开菜单，选择命令，这对用户来说，会觉得比较麻烦。

而快速菜单并不需要到窗口顶部来打开它。只要单击一下鼠标右钮，就会弹出一菜单。这一菜单在设置时，用户可以随意的把它放在某一位置。或者鼠标指针在某一位置，单击时它就出现在那一位置。

不管是设置的隐藏菜单（如前面设计的 Do 菜单，在程序运行时，它并不在窗口上出现，连菜单条也没有），还是在窗口上部显示菜单条的菜单（如 Edit 菜单）。只要我们把它们设置成快速菜单，它们就可以在单击鼠标右钮时在窗口上显示出来。

下面，我们就把前面设计的隐藏菜单 Do 菜单在程序运行时，作为快速菜单显示出来。

首先，双击 Form 窗口，则自动打开 Code 窗口。也可以单击 Project 窗口中的 View Code 按钮进入 Code 窗口。

在 Object 列表框中选则 Form，在 Procedure 列表框中选择 MouseUp，则在 Code 窗口中出现 Form _ MouseUp 事件的过程头与过程尾。

```
Private Sub Form _ MouseUp (Button As Integer,Shift As Integer,
                                    X As Single,Y As Single)
End Sub
```

在事件的过程头与过程尾之间嵌入代码：

```
If Button = 2 Then
  PopupMenu MnuDo
End If
```

这段代码的意思是，当单击鼠标右钮（Button＝2）时，即弹出 Do 菜单。

选择 Visual Basic 中 Run 菜单上的 Start 命令，或单击菜单条的 Start 按钮，运行 myfirst 应用程序。在程序运行时的 Form 窗口中单击鼠标右钮，即弹出 Do 菜单，如图7.14所示。不过，读者会看到 Do 菜单名并未出现。在这里，当在 data1等文本框中单击鼠标右钮时，也会出现一菜单，但这一菜单并不是我们所设计的，而是 Visual Basic 为控件设计的。

在菜单上有菜单名的菜单也可以以快速菜单的形式出现。例如，把 Edit 菜单设计成快速菜单，只要把上面的 FormMouse _ Up 事件过程代码中的 MnuDo 变为 MnuEdit 就可以了。

当程序运行时，单击鼠标右钮，就会出现 Edit 菜单。如图7.15所示。只是菜单上部没有菜单名 Edit。选择其中的某一菜单项，即可执行这一命令。

一般来说，用户应将经常使用的命令如 Close、Clear 等命令组成一新菜单，设计成快速菜单，便于使用。

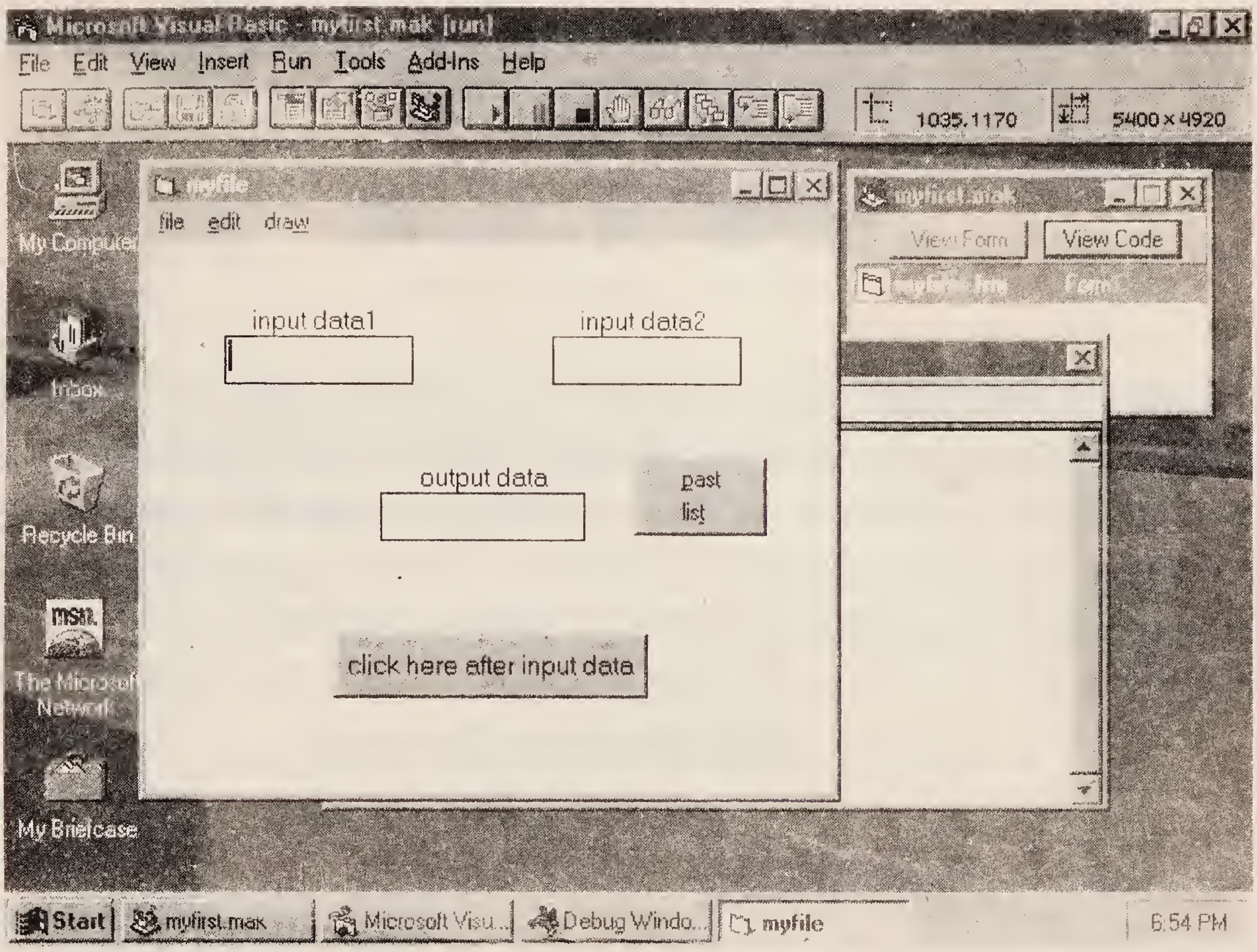

Microsoft Visual Basic - myfirst.mak [run]
File Edit View Insert Run Tools Add-Ins Help
1035,1170
5400 x 4920
myfile
file edit draw
input data1
input data2
output data
past
list
click here after input data
myfirst.mak
View Form
View Code
Start
myfirst.mak
Microsoft Visu...
Debug Windo...
myfile
6:54 PM

图7. 14　Do 快速菜单

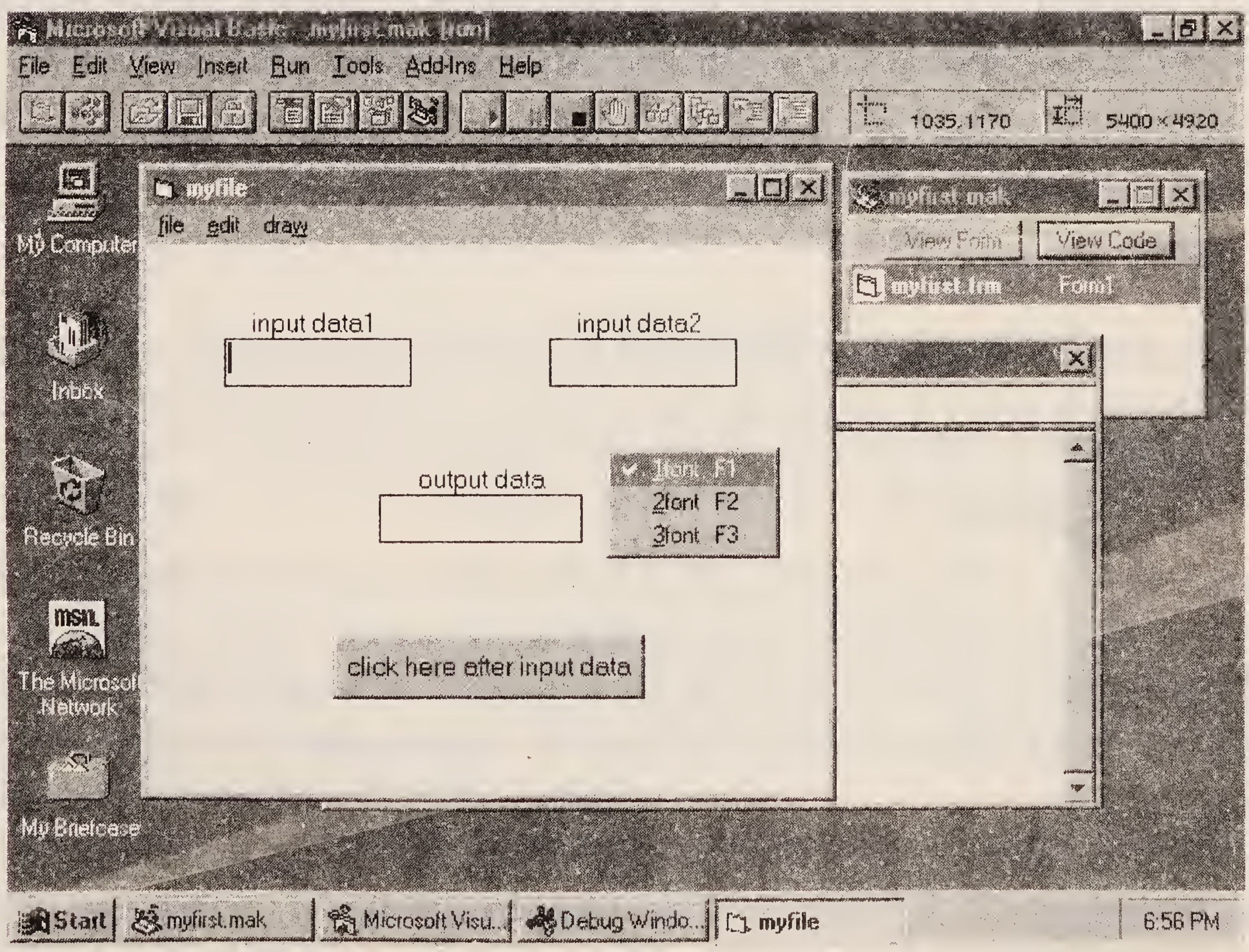

Microsoft Visual Basic - myfirst.mak [run]
File Edit View Insert Run Tools Add-Ins Help
1035,1170
5400 x 4920
myfile
file edit draw
input data1
input data2
output data
✓ 1font F1
2font F2
3font F3
click here after input data
myfirst.mak
View Form
View Code
myfirst.frm Form1
Start
myfirst.mak
Microsoft Visu...
Debug Windo...
myfile
6:56 PM

图7. 15　Edit 快速菜单

第八章　对　话　框

本章介绍的对话框，其实用户已经很熟悉，只是没有意识到，例如前面讲到的存储用户工作所用的Save Project、Save File等菜单选项时都用到了对话框。对话框是在Windows应用程序中经常使用的一类特殊窗体，在Visual Basic中一个对话框同窗体其实并没有什么区别，尽管对话框也许有不同的边框风格或其它属性，但它仍然是包含有控件的窗体。本章主要介绍建立对话框时经常使用到的控件，以及如何建立一个对话框。

8.1　创建对话框的基本知识

8.1.1　怎样定制一个对话框的界面风格

虽然对话框和其它窗体一样，但对话框仍然具有区别于其它窗体的不同属性。在大多数的情况下，用户没有必要改变对话框的大小，所以一般来说，对话框的边框大多都是被认为固定的边框，同时对话框的Min Button和Max Button属性应设成False，如果不被设成False，尽管对话框不会显示最大化按钮和最小化按钮，但是控制菜单中仍然包含有Maxinize和Minimize选项，除非用户把对话框窗体的Control Box属性设置成False。

8.1.2　调入和显示对话框

对话框就像其它窗体一样，使用前必须调入。用户可以通过显式地调入其窗体来调入对话框或者隐含的显示调入的对话框。用户调入和显示一个窗体时经常要用到Show方法，其语法格式如下：

```
[from.]Show[Style]
```

语法中的form是要显示的窗体，如果没有指定，则显示与当前窗体模块相联的窗体。语法中的Style确定窗体是模态还是非模态的整数值，若Style是0，窗体是非模态的，若Style是1，则窗体是模态的。如果一个窗体是模态窗体且被显示，任何程序的其它输入都不会发生直到模态窗体被隐含或释放，但其它的运行程序不受影响。例如，用户可通过下列语句指定一个窗体风格参数：

```
frmDly.Show 1
```

显示非模态窗体时，在Show方法后所遇到的Visual Basic代码将被执行；显示模态窗体时，不会执行Show方法后的Visual Basic代码，直到窗体被隐藏或卸下。在调用Show方法时，若指定的窗体还没有被装入，Visual Basic会自动装入。最常用到的模态对话框有输入框和消息框。例如用下列代码显示一个隐藏窗体。

```
Sub Form _ Click()
    Dim Msg              'Declare Variable
    Hide                 'Hide form
    Msg="Choose ok to make the form reappear"
    MsgRox Msg           'Display message
    Show                 'Show form again
End Sub
```

除了显示一个对话框，还需要决定用户结束使用时，怎样隐藏或释放对话框。用户单击OK按钮并不能自身隐藏窗体，必须向Click事件过程增加一条语句使按钮能隐藏或释放窗体，这就要用到Hide方法，该方法的用法如下：

```
[form]. Hide
```

如果Hide方法被激活而窗体没有装入，则窗体被装入但不显示。Hide方法并不把窗体从内存中删去，如果要把窗体从内存中删去，必须用Unload语句。下面两条语句使对话框从显示器上删去。

```
frmDly. Hide
Unload frmDly
```

8.1.3 为对话框的控件设置属性

用户可以在设计时为控件设置有关属性，但在有些情况下，用户必须在运行时设置控件的属性，因为这些设置是基于程序中的条件判决。即使对话框被隐藏，用户也可以设置其控件的属性，但要记住除了指定控件名还要指定窗体名，例如：

```
frmDly! Cmd open. Value=True
```

只要指定窗体名、惊叹号、控件名、属性和属性值即可。

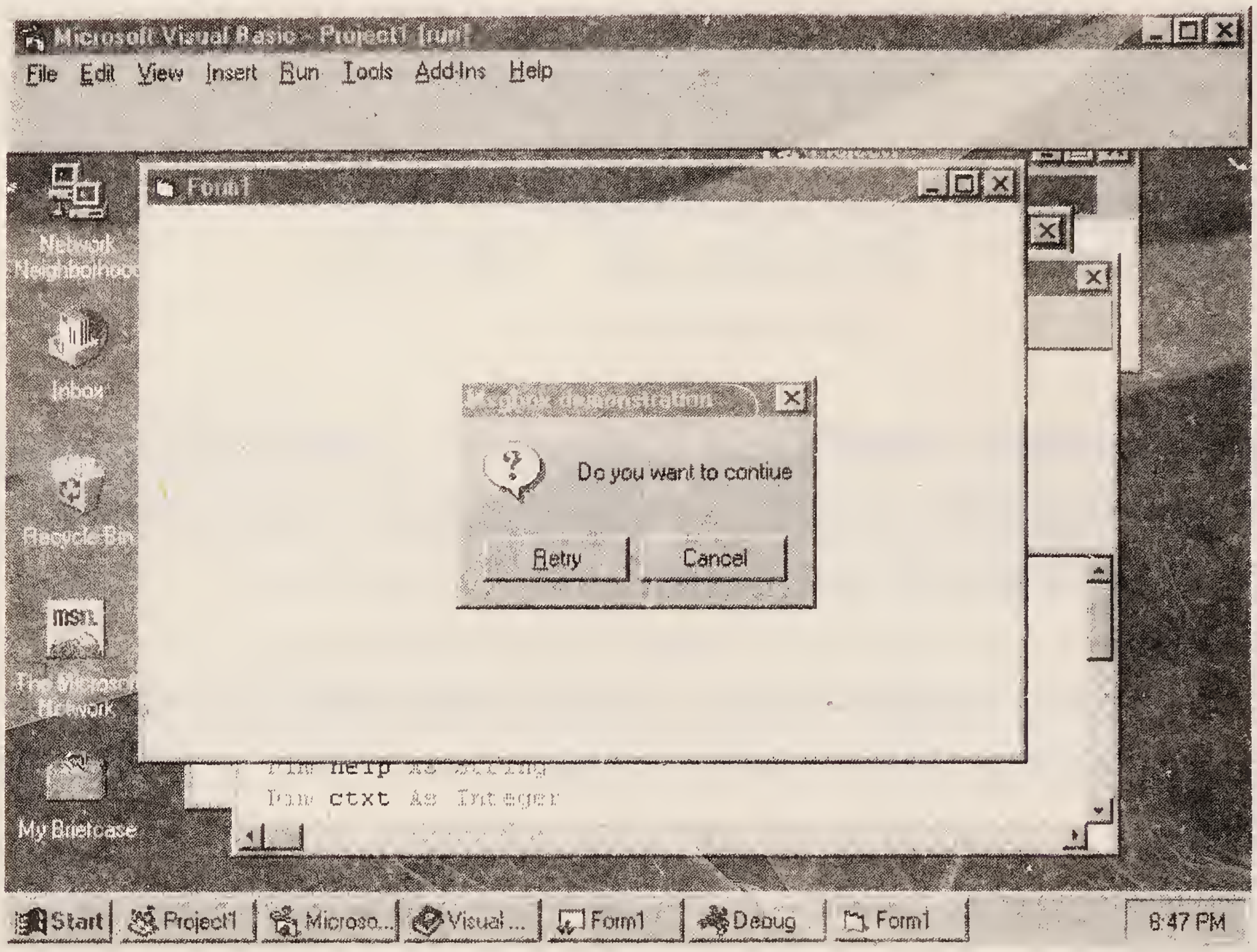
8.2 预制对话框

在Visual Basic中对话框有三种类型：预制对话框、定制对话框和公用对话框，本节主要讲预制对话框的建立。Visual Basic能够建立两种预制对话框：消息框和输入框。

8.2.1 消息框

可能用户经常遇见消息框，每当正在使用的程序出了问题，屏幕就会出现一个小对话框，告诉用户程序出现了问题，选择OK按钮结束，消息框为用户提供了一个快捷的显示有关信息的对话框，而不必为对话框画一个窗体。

MsgBox函数让用户在一个简单的对话框中显示消息，这个对话框可包括预定义的按钮和项目，用户要指定对话框中所使用的消息、标题、按钮和项目，图8.1就是由MsgBox函数产生的一个典型的对话框。


Microsoft Visual Basic - Project1 [run]
File Edit View Insert Run Tools Add-Ins Help
Form1
Msgbox demonstration
Retry
Cancel
Network Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Start
Project1
Microso...
Visual ...
Form1
Debug
Form1
8:47 PM

Do you want to contiue

```
Dim ctxt As Integer
```

图8.1 MsgBox 函数产生的对话框

1. MsgBox 函数

该函数能够在对话框中显示信息和等待用户选择按钮。MsgBox 函数描述用户已选择的按钮。该函数的语法格式如下：

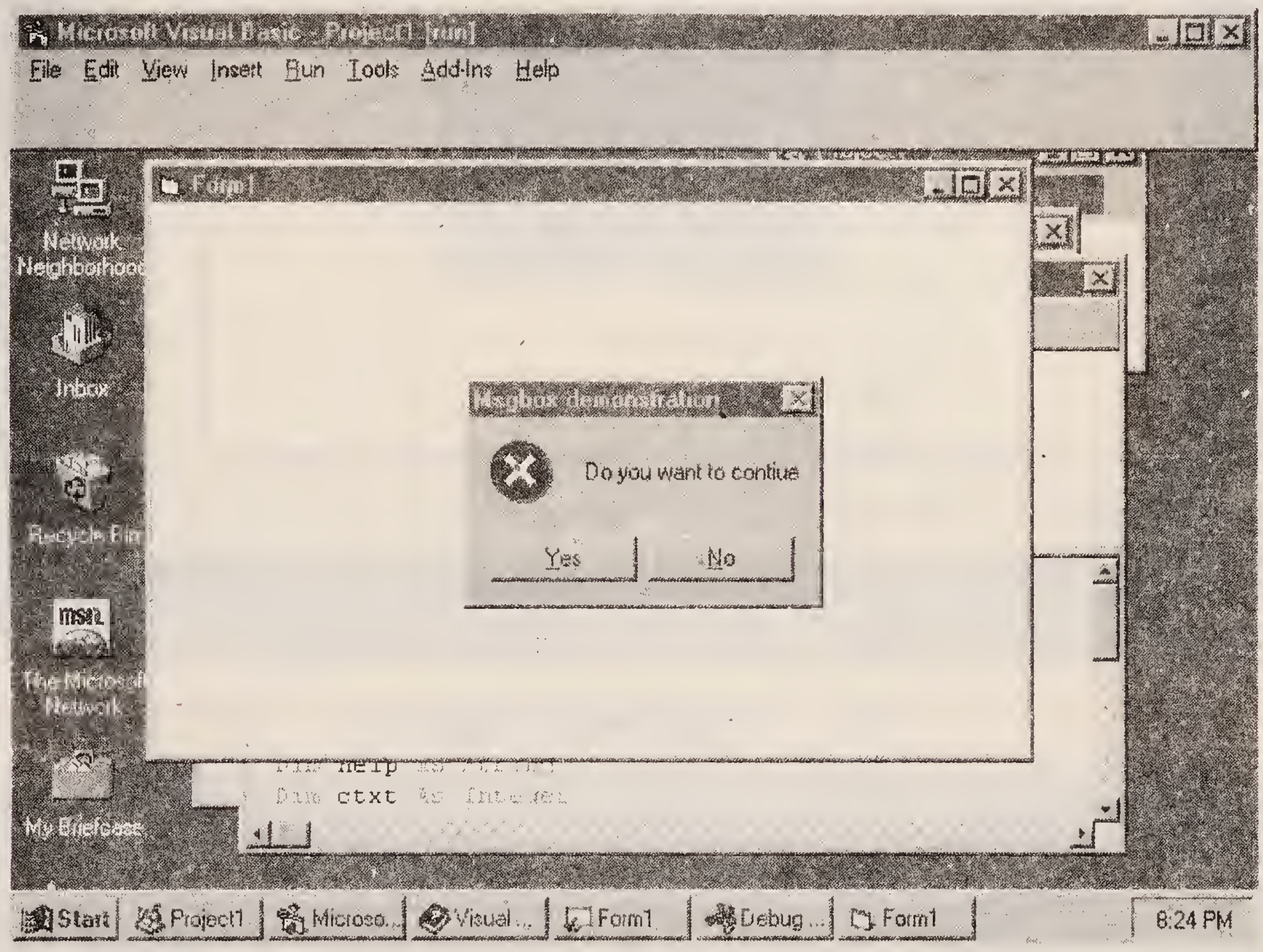
MsgBox(prompt[,buttons][title][,helpfile,context])

其中 prompt 代表对话框中作为信息显示的字符串表达式，字符串的最大长度为1024个字符。title 参数代表对话框中标题条显示的字符串表达式，如果省略 title 参数，则 MsgBox 函数用“Microsoft Visual Basic”作为缺省标题。buttons 代表可选的数值表达式，是指定显示按钮号和类型值、使用的图标类型以及缺省按钮和模式值的和。下面的表格说明了数字的使用和每组数值的含义：

符号常量	值	描述
VbOKonly	0	只显示 OK 按钮
VbOKCancel	1	显示 OK 和 Cancel 按钮
VbAbortRetryIgnore	2	显示 Abot、Retry 和 Ignore 按钮
VbYesNoCancel	3	显示 Yes、No 和 Cancel 按钮
VbYesNo	4	显示 Yes 和 No 按钮
VbRetryCancel	5	显示 Retry 和 Cancel 按钮
VbCritical	16	显示图8.2中所示的图标
VbQuestion	32	显示图8.3中所示的图标
VbExclumation	48	显示图8.4中所示的图标
VbInformation	64	显示图8.5中所示的图标
VbDefaulButton1	0	第一个按钮缺省
VbDefaulButton2	256	第二个按钮缺省
VbDefaulButton3	512	第三个按钮缺省

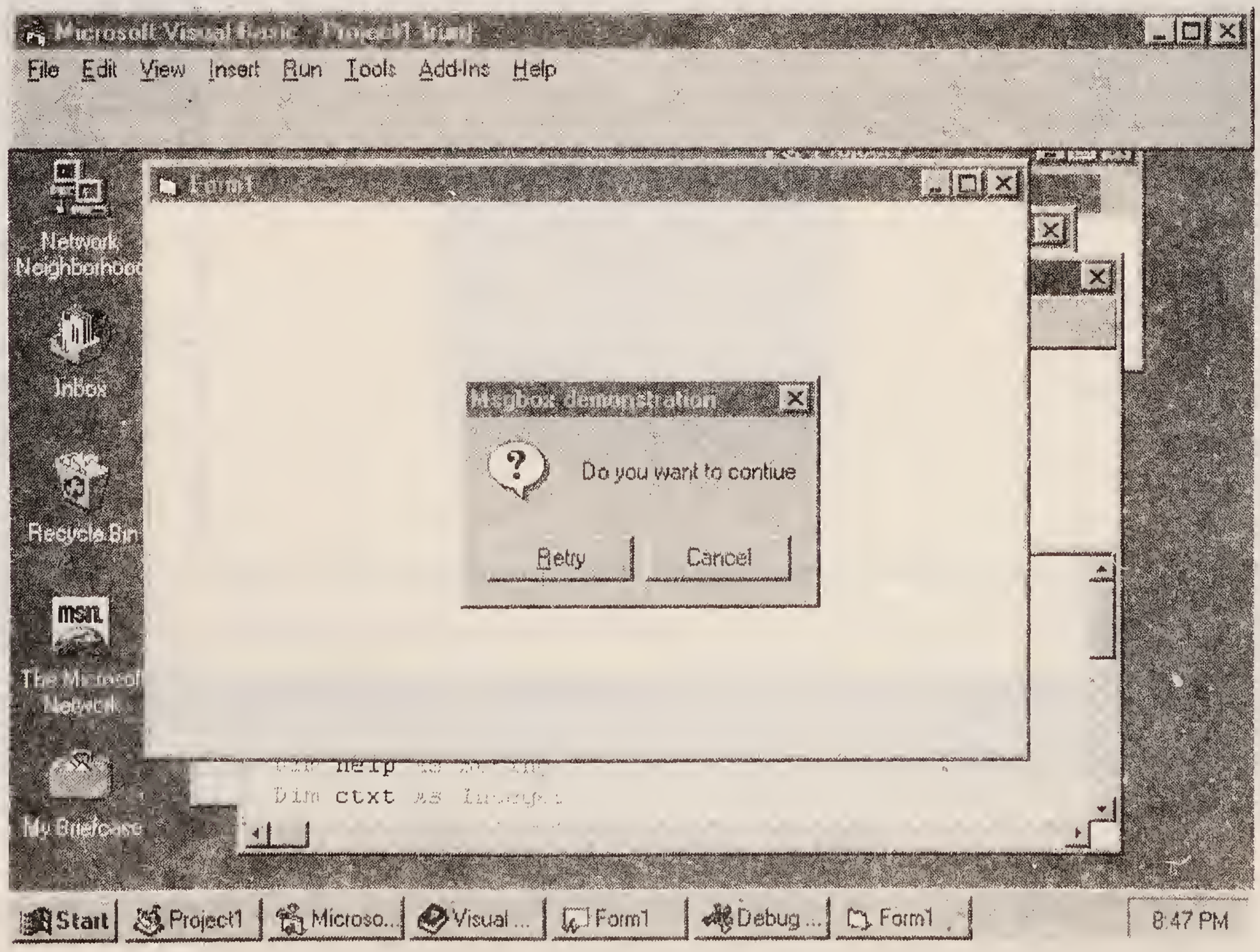
（续表）

符号常量	值	描述
VbApplicationModel	0	应用模式对话框，用户必须在当前应用程序上工作之前对信息框做出反应
VbSystemModel	4096	系统模式对话框，所有应用程序在用户回应消息之前处于暂停状态


Microsoft Visual Basic - Project1 [run]
File Edit View Insert Run Tools Add-Ins Help
Form1
Network Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Msgbox demonstration
Yes
No
Start
Project1
Microso...
Visual ...
Form1
Debug ...
Form1
8:24 PM

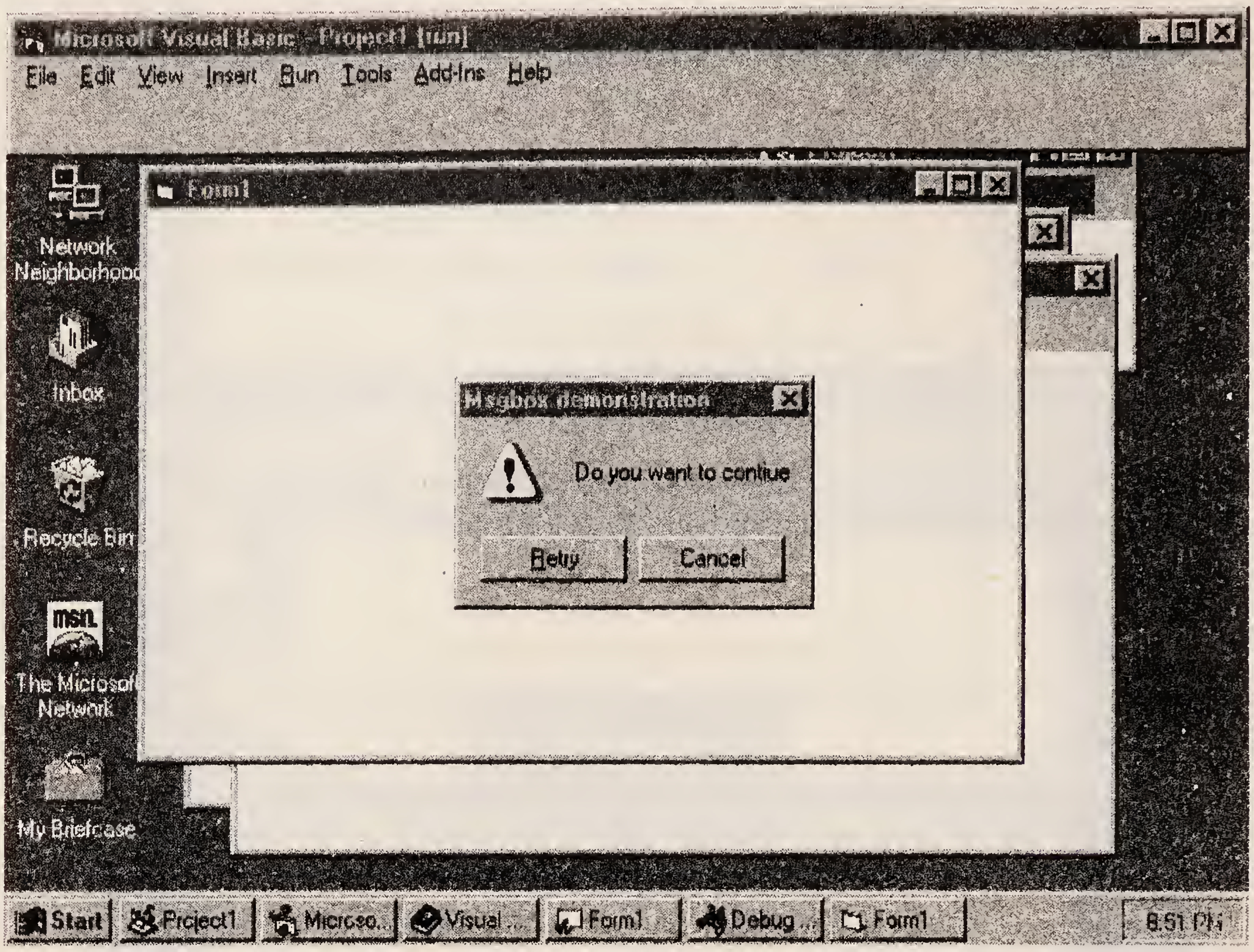
Do you want to contiue

图8.2 消息框中显示的图标


Microsoft Visual Basic - Project1 [run]
File Edit View Insert Run Tools Add-Ins Help
Form1
Network Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Msgbox demonstration
Retry
Cancel
Start
Project1
Microso...
Visual ...
Form1
Debug ...
Form1
8:47 PM

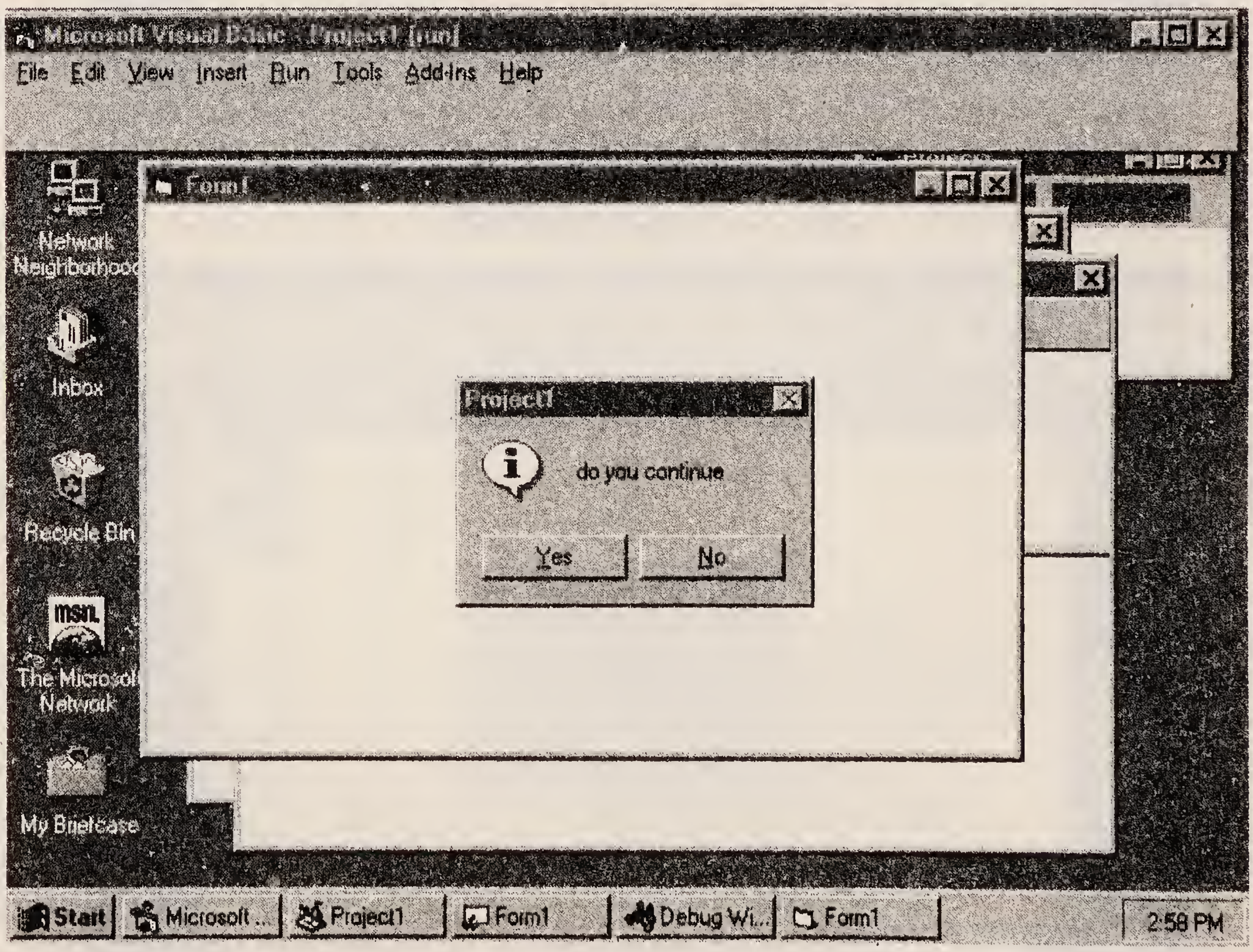
Do you want to contiue

图8.3 消息框中显示的图标


Microsoft Visual Basic - Project1 [run]
File Edit View Insert Run Tools Add-Ins Help
Form1
Network Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Msgbox demonstration
Retry
Cancel
Start
Project1
Microso...
Visual ...
Form1
Debug ...
Form1
8:51 PM

Do you want to contiue

图8.4　消息框中显示的图标


Microsoft Visual Basic - Project1 [run]
File Edit View Insert Run Tools Add-Ins Help
Form1
Network Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Project1
Yes
No
Start
Microsoft ...
Project1
Form1
Debug Wi...
Form1
2:58 PM

do you continue

图8.5　消息框中显示的图标

注意　上表第一组数值(1—5)描述对话框中显示按钮的数字和类型；第二组(16,32,48,64)描述图标类型；第三组数(0,256,512)决定哪一个按钮缺省；第四组(0,4096)决定消息框模式。当用户增加值来产生 buttons 参数的最后数值时，只能够从每组用一个数，如果

省略，则缺省值为零。

语法格式的 helpfile 参数用来指定为消息框提供主题帮助的帮助文件。Context 参数表示一个数值表达式，指定帮助目录数用以帮助用户安排适当的帮助主题，Context 参数和 Helpfile 参数必须同时被提供。当 helpfile 和 Context 两种参数都被提供时，用户能通过按 F1 键观看同帮助目录相关的帮助主题。

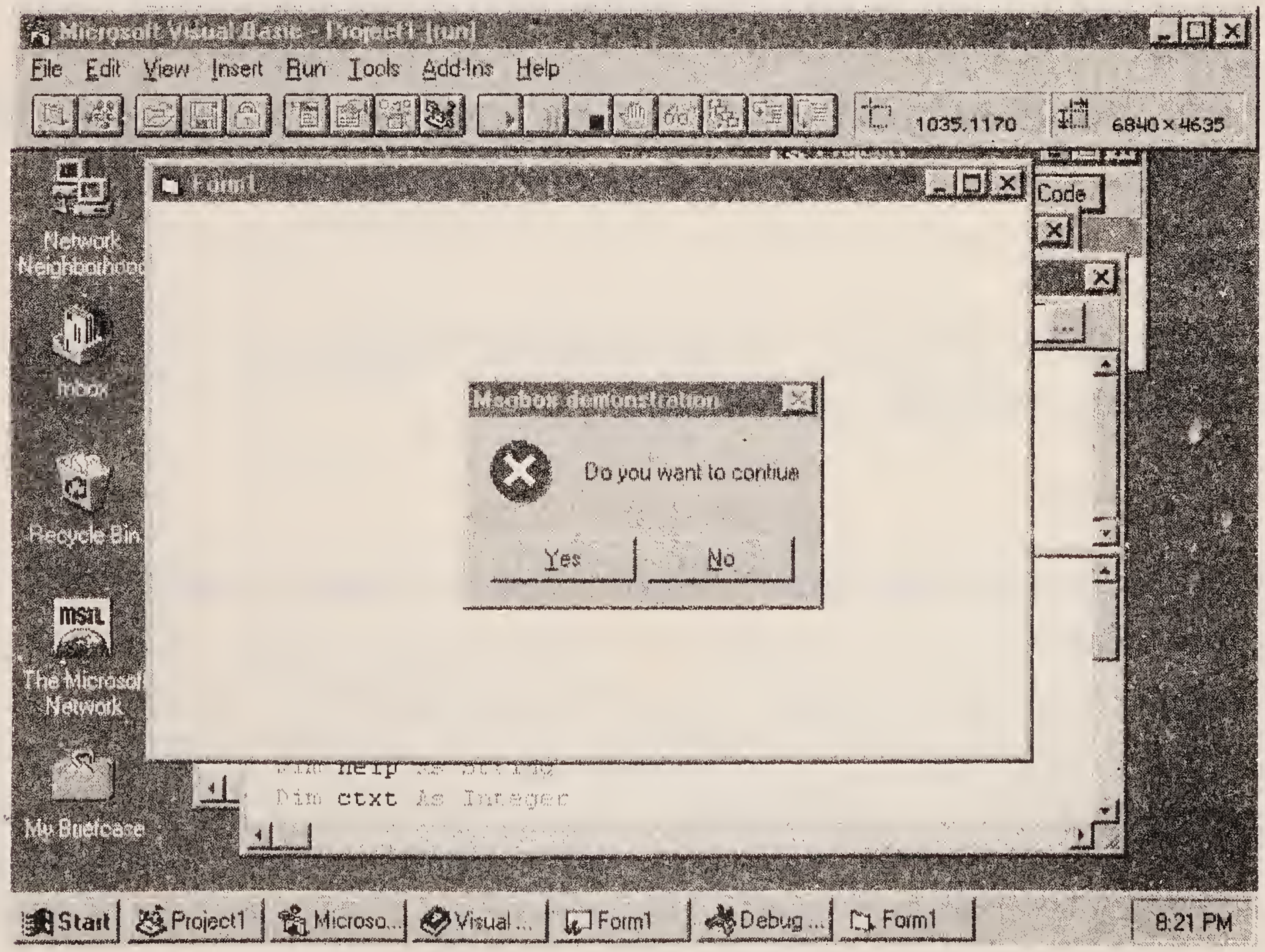
MsgBox 函数还可以通过返回值表明选择了哪一个按钮，对应情况如下表。

符号常量	数值	含义
VbOk	1	选择 OK 按钮
VbCancel	2	选择 Cancel 按钮
VbAbort	3	选择 Abort 按钮
VbRetry	4	选择 Retry 按钮
VbIgnore	5	选择 Ignore 按钮
VbYes	6	选择 Yes 按钮
VbNo	7	选择 No 按钮

注意 对话框中显示一个 Cancel 按钮，按下 ESC 键与选择 Cancel 起相同的作用。

2. 建立消息框

如果用户想建立如图8.6所示的消息框来询问用户是否继续工作，可以按下列步骤进行。



Microsoft Visual Basic - Project1 [run]
File Edit View Insert Run Tools Add-Ins Help
1035,1170
6840×4635
Form1
Code
Network Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Msgbox demonstration
Yes
No
Start
Project1
Microso...
Visual ...
Form1
Debug ...
Form1
8:21 PM

Do you want to contiue

```
Dim help As String
Dim ctxt As Integer
```

图8.6 消息框

(1) 设置消息框的标题为“Msgbox demonstration”。

(2) 设置消息框显示的信息为“Do you want to continue”。

(3) 选择按钮的类型为 VbYesNo。

(4) 选择图标和缺省按钮分别为 Vbcritical 和 VbDefaultButton2。

根据以上步骤可以编写以下代码：

```
Sub Form _Click()
Dim msg As String
Dim style As Integer
Dim title As String
Dim response As Integer
Dim MyString As String
Msg="Do you want to continue"
Style=VbYesNo + VbCritical + VbDefaultButton2
title="Msgbox demonstration"
response=MsgBox(msg,style,title)
If response = VbYes Then
          MyString = "Yes"
   Else
          MyString = "No"
   End If
End Sub
```

用户把以上代码键入单击事件过程中，当单击窗体中的空白处就会弹出如图8.6所示的对话框。

8.2.2　输入框

输入框和消息框服务的目的不一样，输入框是从用户得到信息，在 Visual Basic 4.0中提供了 InputBox 函数用来帮助用户建立输入框。

1. InputBox 函数

该函数能够在对话框中显示提示符，等待用户输入文本或选择一按钮，并且返回文本框的内容，它的语法格式为：

ImputBox(prompt[,title][,default][,Xpos][,Ypos][helpfile,Context])

其中 prompt 字符串表达式作为对话框的信息显示，提示符的最大长度约为1024个字符(取决于使用字符的宽度)，如果提示符超过一行，用户能够在每行之间用回车(ANSI 字符13)和一个换行符(ANSI 字符为10)来分隔每行。title 字符串表达式作为对话框的标题显示，如果用户忽略设置 title，则应用程序名被放在标题条作为输入框的标题名。如果用户没有提供其它输入，default 字符串表达式作为缺省值在文本框中显示，如果省去 default，则文本框显示空白。Xpos 数值表达式用点指示从屏幕左端到对话左端的水平距离，若 Xpos 省去，对话框水平居中。Yops 数值表达式用点指示从屏幕上端到对话上端的垂直距离，若 Ypos 被省去，对话垂直位于屏幕下方大约三分之一处，若省略 Xpos，也必须省略 Ypos。helpfile 字符串

表达式作为提供给对话框目录帮助方式的帮助文件。Context 数值表达式作为安排给相关帮助主题的帮助目录数。helpfile 和 Context 两者必须同时被提供。当 helpfile 和 Context 被提供，帮助按钮自动增加到对话框中。如果用户选择 OK 按钮或按下回车键，则 InputBox 函数返回文本框中所有内容。如果用户选择 Cancel 按钮，则函数返回零长度字符串。

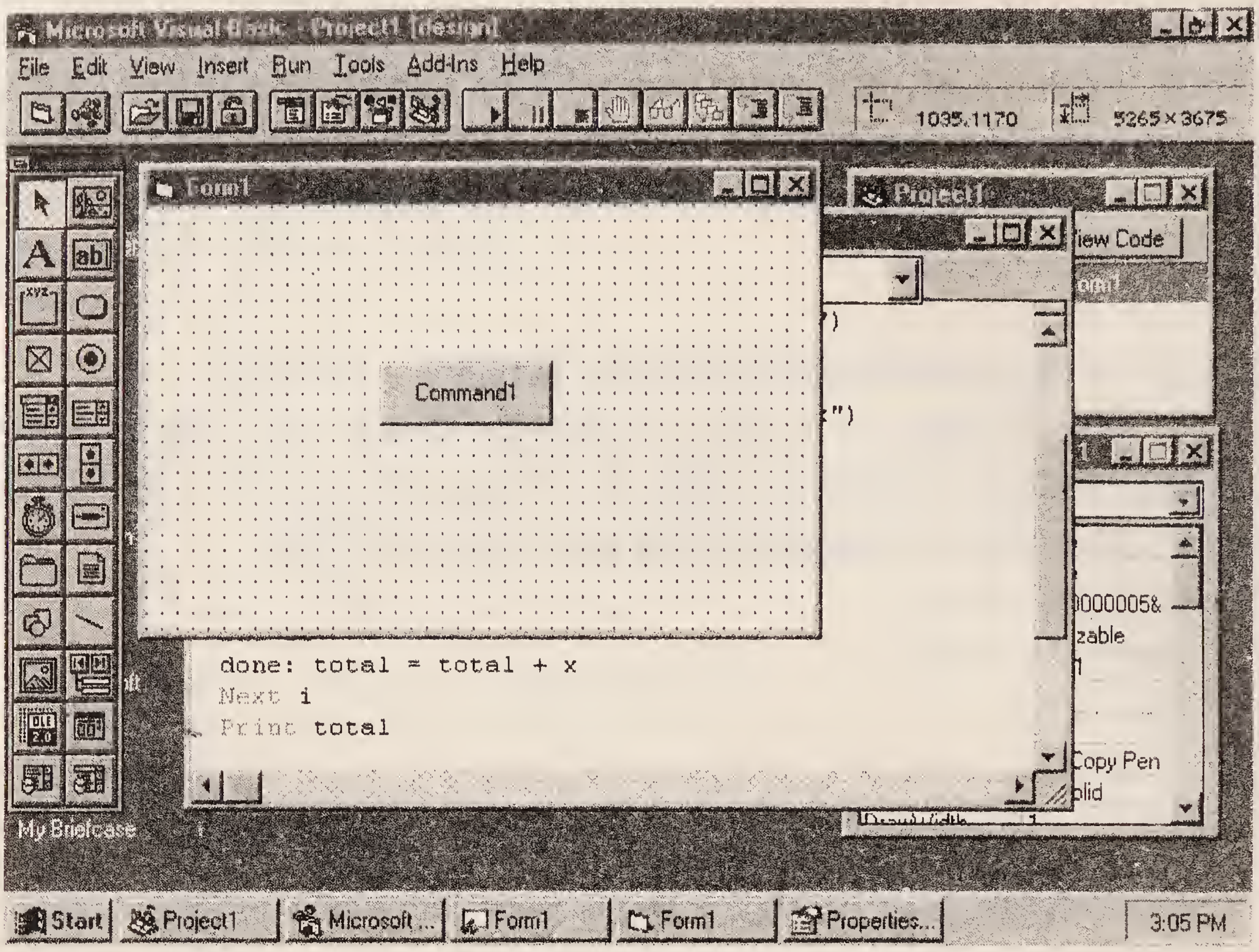
2. 建立输入框

下面将用一个例子来说明如何利用 InputBox 函数建立一个输入框。

例：设计一程序，可以输入5个介于60－100间的数并做累加工作，如果所输入的数超出此范围，则先跳出来问用户是否要输入此数，用户若从消息框中选“Yes”，则此数当作有效输入继续执行，若用户选"No"则此数当作无效输入。下面是本例代码：

```
Sub Commond1 _Click()
    Commond1. Visible=0
    For i=1 to s
        again:
            msg1 $ ="enter number 60-100"
            msg2 $ ="input this number?"
            spec $ =InputBox(mys1 $ , "Inputbox","0")
            X=Val(Spec $ )
    If X<60 or X>100 Then
        Check = MsgBox(msg2 $ r,36,"Check box")
            If Check = 6 Then
                GoTo done
            ElseIf Check = 7 Then
                GoTo again
            End If
    End If
      done: total=total+X
            Next i
            print total
End Sub
```

上述代码中，第一个语句（Command1. Visible＝0）用来使窗体上的命令按钮消失不见，第10行使用 MsgBox 函数来传送信息（其中有2个参数省略），当用户输入超出范围 X＜60或 X＞100时，此语句中 Check 变量用来存放 MsgBox 函数的返回值（由用户选某一按钮决定）。若返回是6，表示用户选“Yes”，此输入值被认为有效，所以转入 done 继续累加工作。若返回值是7，表示用户选择“No”，此值被认为无效，程序转入 again：行重新输入，用户单击 Command1按钮，见图8. 7，窗体中弹出如图8. 8所示的输入框，输入框中文本框缺省值为0，如果用户输入60－100间的数，再单击 OK 又重新出现输入框要求用户再次输入，如果用户输入的数据超出60－100的范围，单击 OK 时，则出现图8. 9的消息框询问用户是否输入这个数。最后在屏幕上输出结果。


Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
1035,1170
5265×3675
Form1
Command1
Project1
View Code
Start
Project1
Microsoft ...
Form1
Form1
Properties...
3:05 PM

```
done: total = total + x
Next i
Print total
```

图8.7　命令按钮

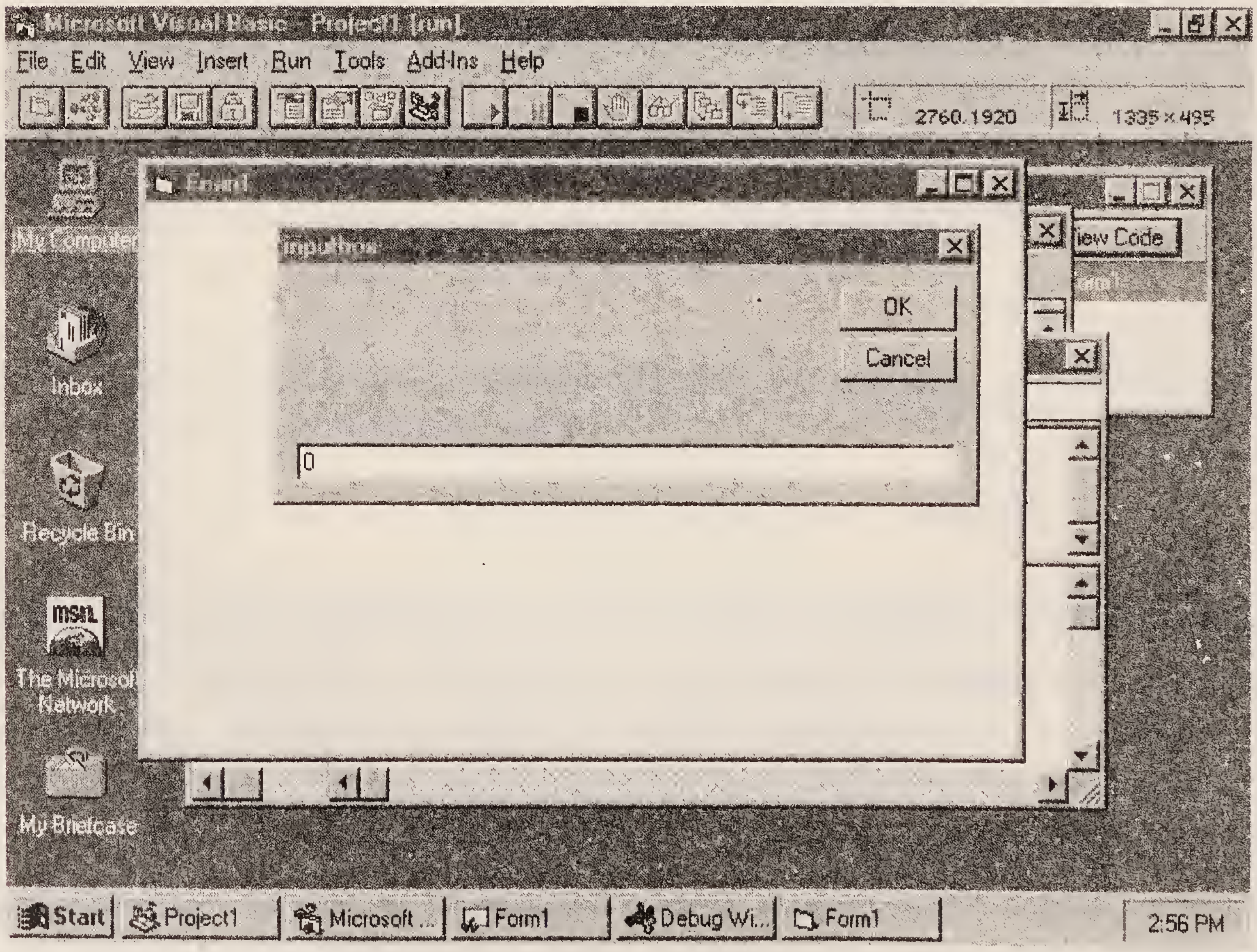
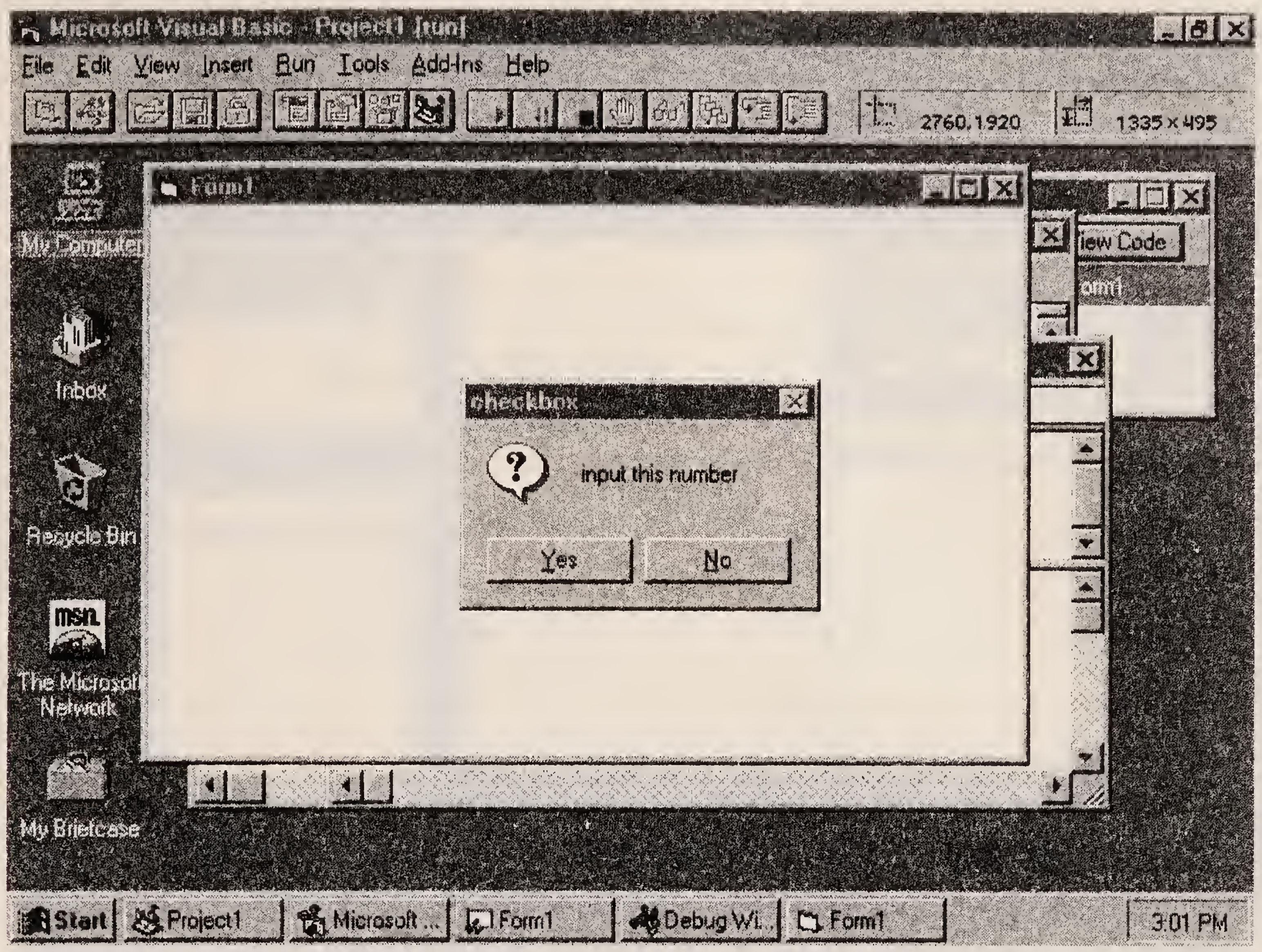

Microsoft Visual Basic - Project1 [run]
File Edit View Insert Run Tools Add-Ins Help
2760,1920
1335×495
Form1
Inputbox
OK
Cancel
0
My Computer
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
View Code
Start
Project1
Microsoft ...
Form1
Debug Wi...
Form1
2:56 PM

图8.8　输入窗框


Microsoft Visual Basic - Project1 [run]
File Edit View Insert Run Tools Add-Ins Help
2760,1920
1335 x 495
Form1
checkbox
Yes
No
View Code
My Computer
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Start
Project1
Microsoft ...
Form1
Debug Wi...
Form1
3:01 PM

input this number

图8.9 确认的消息框

8.3 控件的建立

在对话框中经常要用到像单选按钮、检查框、组合框和列表框等控件，本节主要介绍怎样建立这些控件，它是建立一个对话框的基础。

8.3.1 单选按钮的建立

单选按钮经常在对话框中用到，在许多情况下，至少简单地将单选按钮放在框架里，并通过唯一的名字来引用每一个单选钮，因为单选按钮只可能有两个值—— True 或 False，因此很容易地将单选按钮加入条件语句中。下面将举一例说明怎样建立单选按钮。

例1，本例将建3种控件，一是文本框，另两种是单选按钮和框架，此例的初始画面如图8.10，它们的属性分别叙述如下：

控件	控件名称(Name)	标题(Caption)	文本(Text)
文本框	Screen	没有	"Visual Basic Version 4.0"
框架	Frame1	SelectionOption	没有
单选钮1	modern	fontname(modern)	没有
单选钮2	courier	fontname(Courier)	没有

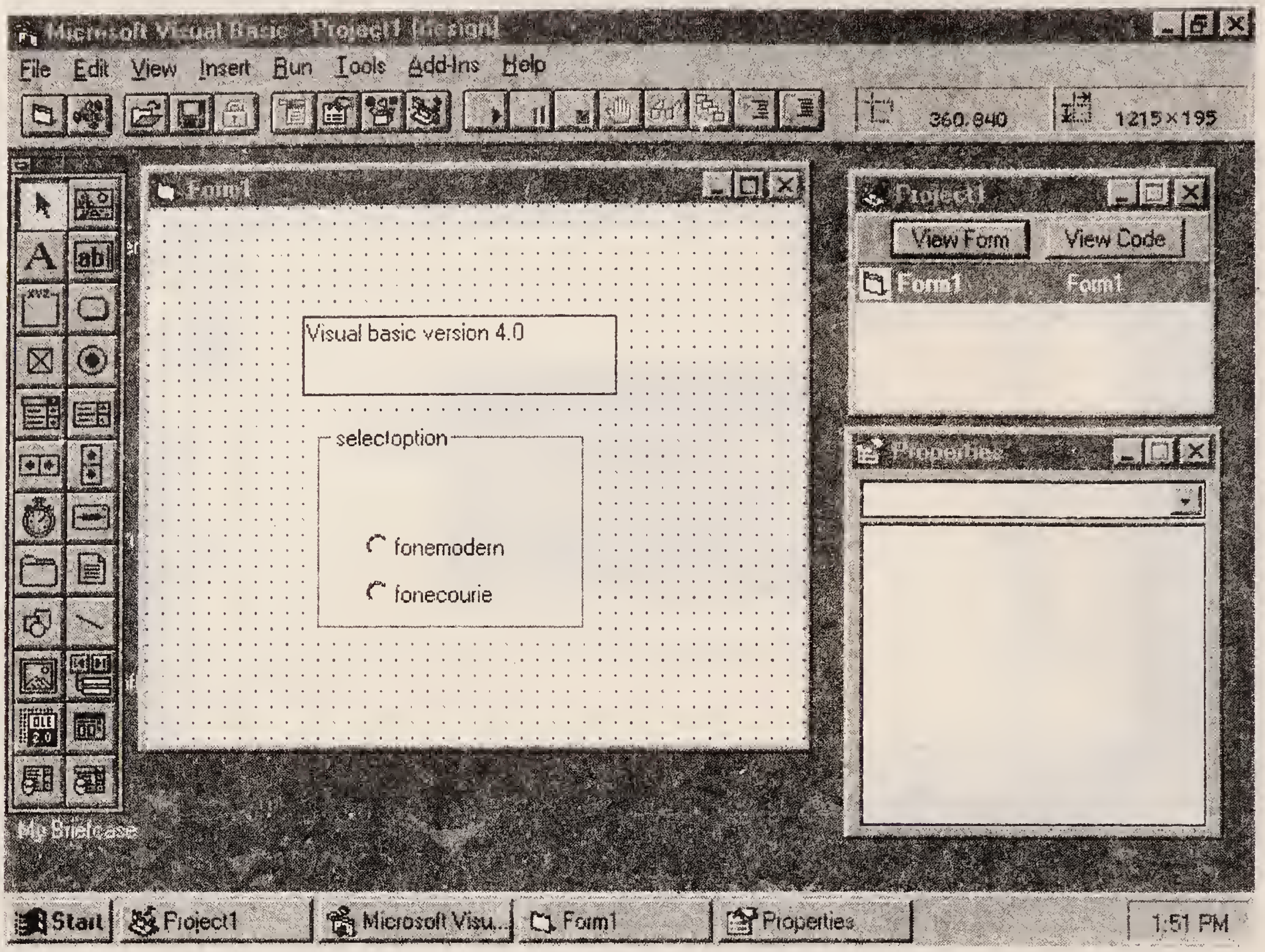

Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
360, 840
1215×195
Form1
Visual basic version 4.0
selectoption
fonemodern
fonecourie
Project1
View Form
View Code
Form1
Form1
Properties
My Briefcase
Start
Project1
Microsoft Visu...
Form1
Properties
1:51 PM

图8.10 初始画面

本例的执行是要使文本框已存在的文本内容经单击“单选钮”来改变显示字体，下面是该例子的事件过程的代码：

```
Sub modern _ Click()
    Screen.fontname="modern"
    Screen.fontname=14
End Sub
Sub courier _ Click()
    Screen.fontname="courier"
    Screen.fontsize=14
End Sub
```

当用户单击 fontmodern 单选钮时，文本框的字体发生变化，见图8.11。

8.3.2 复选框的建立

复选框控件表示某种条件或情况是打开或关闭，它和单选按钮的最大区别是用户可以同时选择多个复选框，我们将举一个较为简便的例子来说明复选框的建立步骤，该例子用复选框来决定文本框显示的字体。

（1）单击 File 菜单的 New Project 命令，并在窗体上建立如图8.12所示的初始界面，包含一个标签、一个文本框和两个复选框。

（2）设置控件的属性如下表所示。

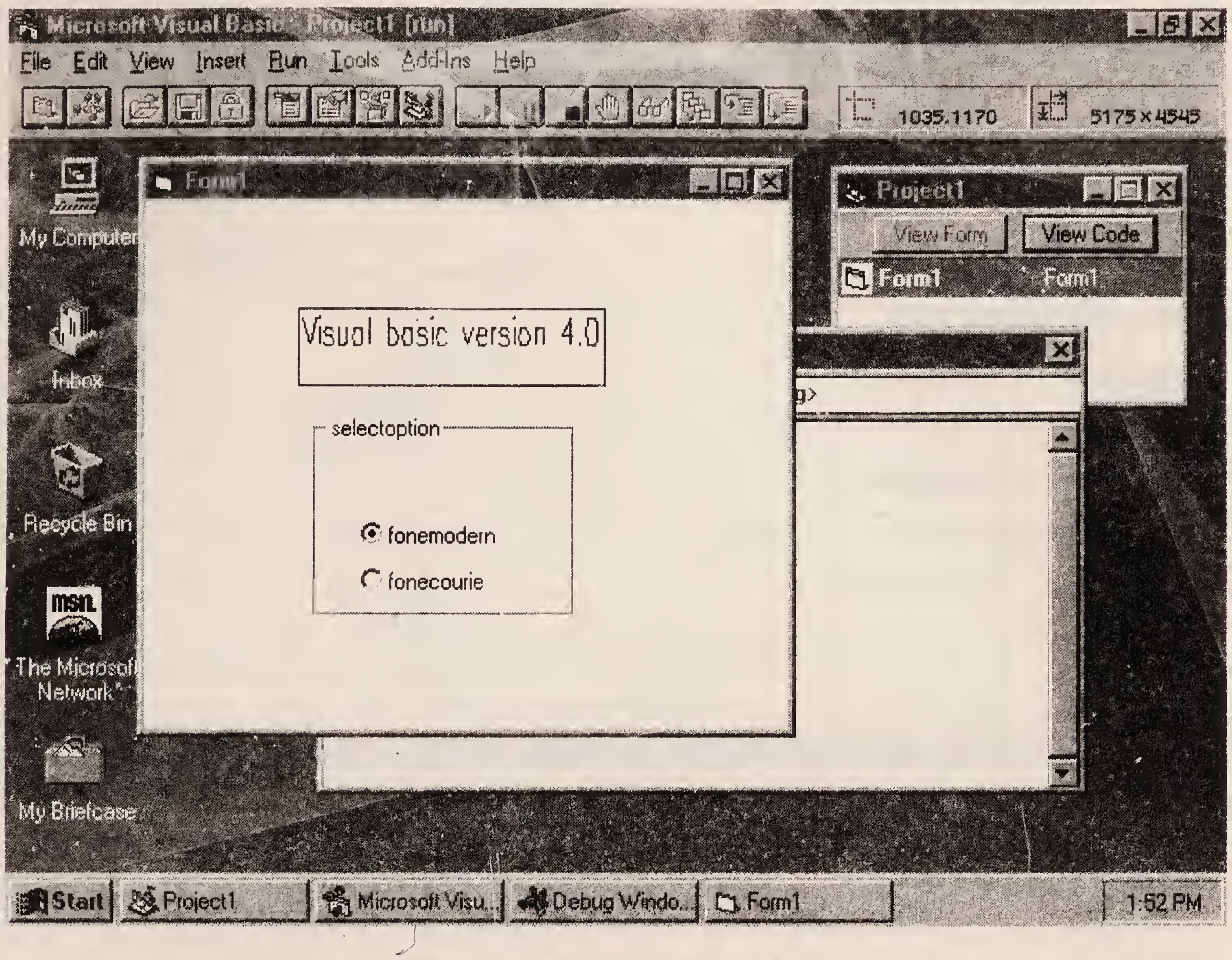

Microsoft Visual Basic - Project1 [run]
File Edit View Insert Run Tools Add-Ins Help
1035,1170
5175 x 4545
Form1
Visuol bosic version 4.0
selectoption
◉ fonemodern
○ fonecourie
Project1
View Form
View Code
Form1 Form1
My Computer
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Start
Project1
Microsoft Visu...
Debug Windo...
Form1
1:52 PM

图8. 11　单击某按钮时的画面

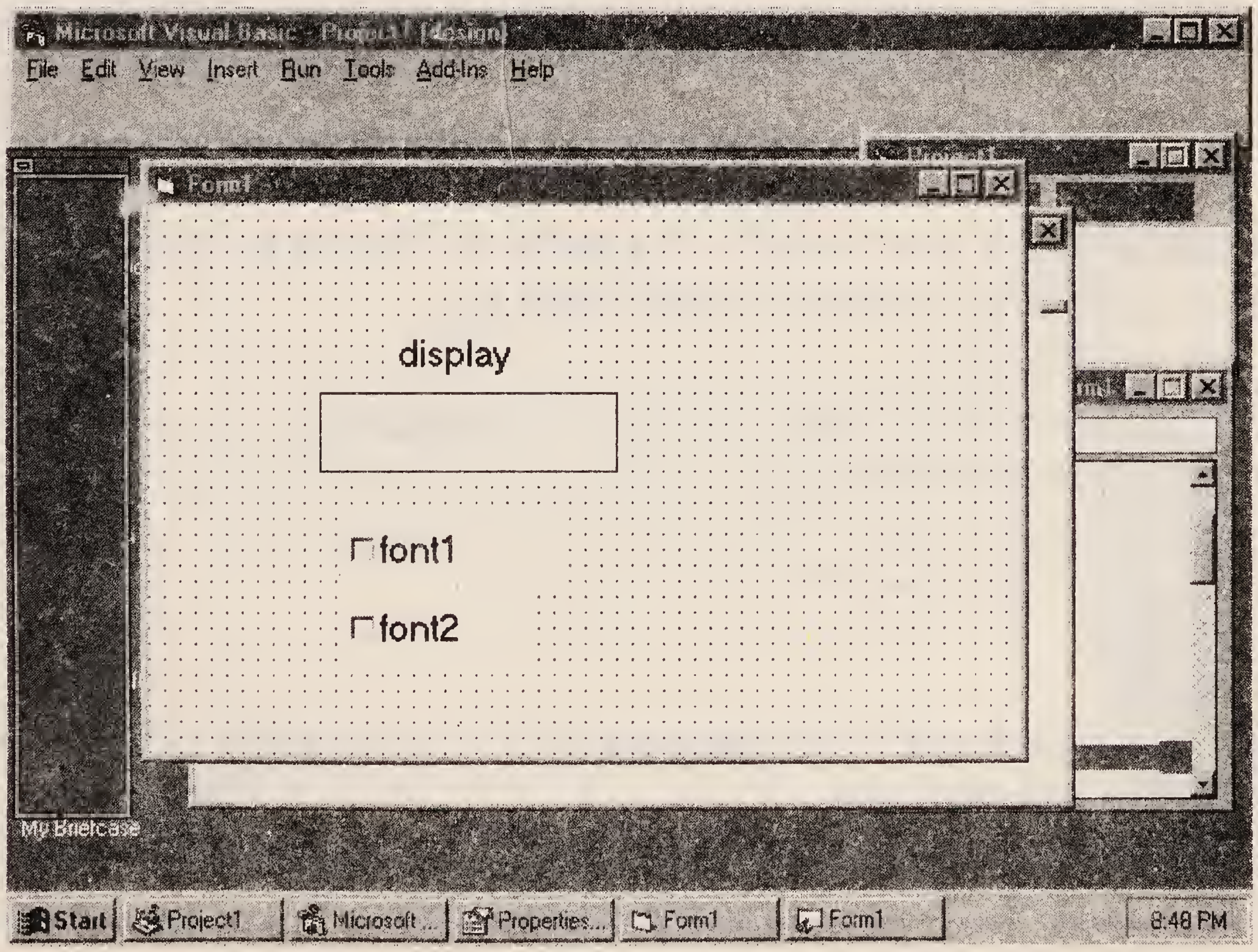

Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
Form1
display
☐ font1
☐ font2
My Briefcase
Start
Project1
Microsoft ...
Properties...
Form1
Form1
8:48 PM

图8. 12　初始界面的建立

控件	控件名称	标题	文本
标签	Label 1	display	没有
文本框	Screen	没有	“ ”
检查框1	Ctlfont 1	font 1	没有
检查框2	Ctlfont 2	font 2	没有

（3）编写各事件的代码。

```
Sub Screen _ Change()
    If Ctlfont1. Value=1 Then
        Screen. fontBold=-1
    ElseIf Ctlfont2. Value = 1 Then
        Screen. fontunderline = -1
    End If
End Sub
```

该事件过程是当用户向文本框中输入时所做出的反应，根据复选框的被选情况来决定文本框中显示的字体。

```
Sub Ctlfont1_Click()
    If Ctlfont1. Value=1 Then
        Screen. fontBold=-1
    Else
        Screen. fontBold=0
    End If
End Sub
```

该单击事件过程是当用户单击第一个复选框时，所产生的反应，它的作用是测试 Ctlfont1. Value 属性从而决定文本框的内容是否以黑体显示。

```
Sub Ctlfont2 _ Click()
    If Ctlfont2. Value=1 Then
        Screen. fontunderline=-1
    Else
        Screen. fontunderline=0
    End If
End Sub
```

该单击事件过程是对第二个复选框所做的反应，它的作用是测试 Ctlfont2. Value 属性从而决定文本框中是否加下划线。

（4）用户键入 F5键运行程序，向文本框输入一字符串，例如“ABCDEF”，见图8.13。然后单击 Font1复选框，字符加粗。单击 Font2复选框，字符被加下划线，见图8.14。

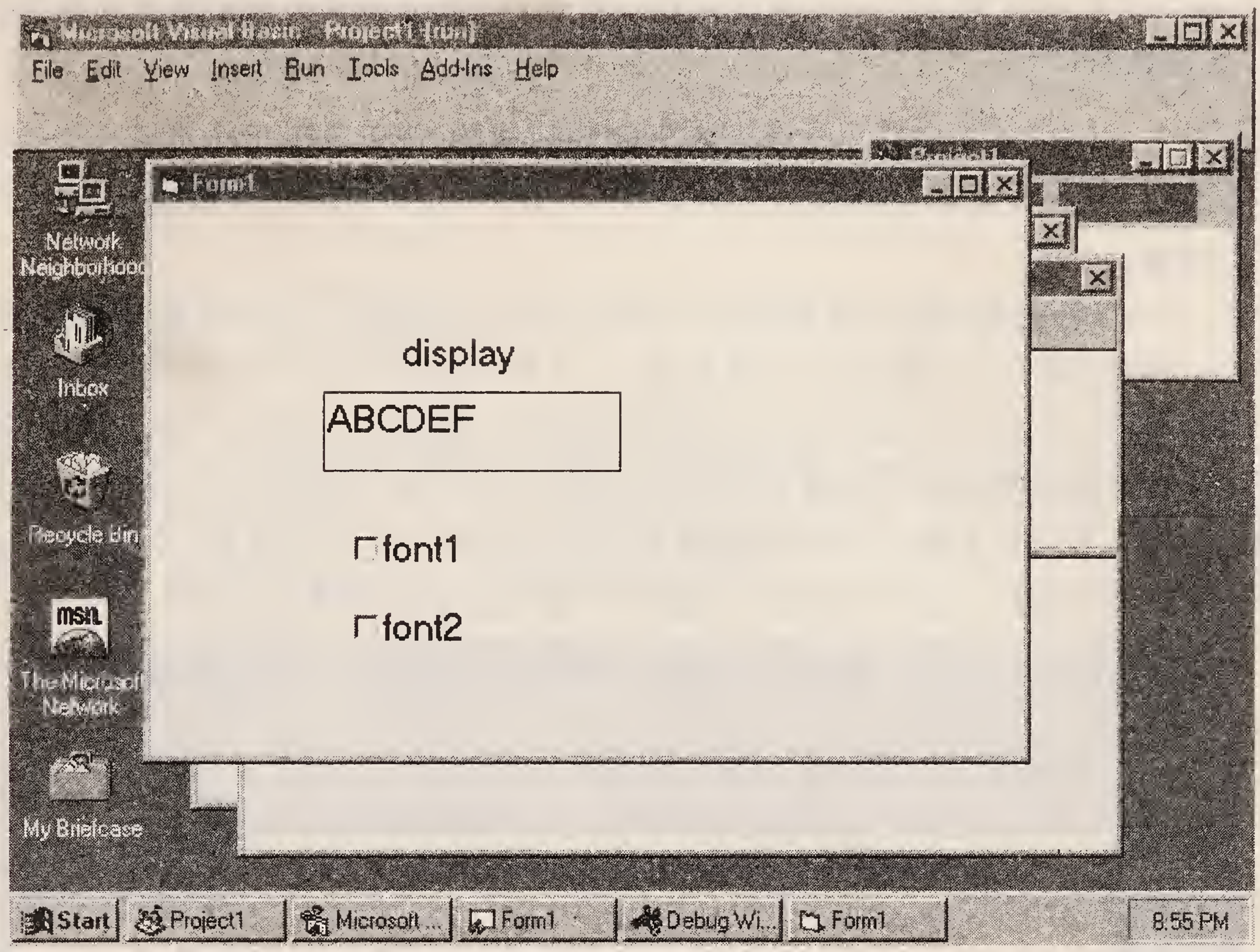

Microsoft Visual Basic - Project1 [run]
File Edit View Insert Run Tools Add-Ins Help
Form1
display
ABCDEF
☐ font1
☐ font2
Network Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Start
Project1
Microsoft ...
Form1
Debug Wi...
Form1
8:55 PM

图8. 13 输入字符串

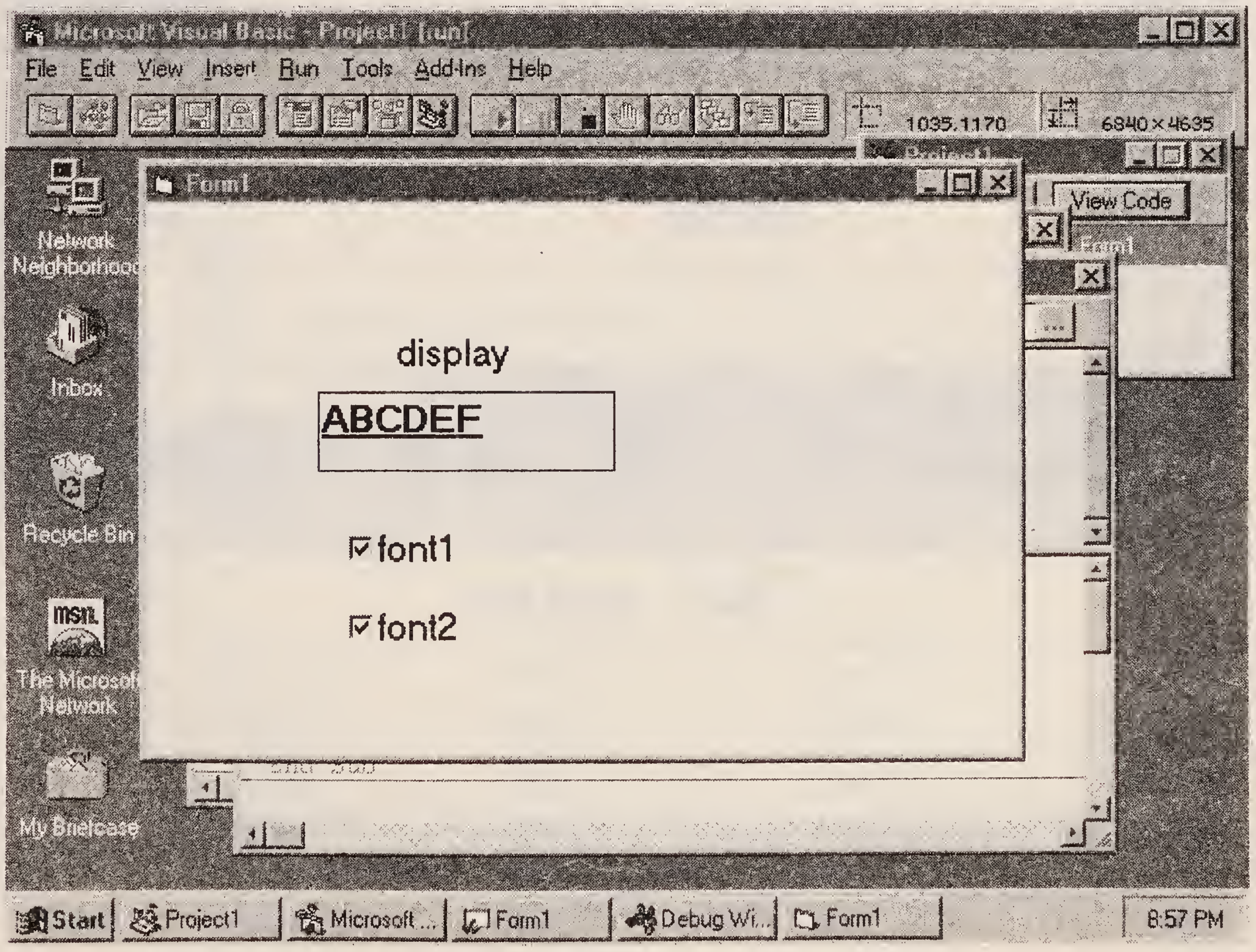

Microsoft Visual Basic - Project1 [run]
File Edit View Insert Run Tools Add-Ins Help
1035,1170
6840×4635
Form1
View Code
display
ABCDEF
☑ font1
☑ font2
Network Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Start
Project1
Microsoft ...
Form1
Debug Wi...
Form1
8:57 PM

图8. 14 字符变化

8.3.3 文本框的建立

文本框是为用户输入或输出文本而提供的一种控件。用户可以用文本框显示信息，也可以向文本框输入信息。文本框的建立过程很简单，该小节主要讲在文本框建立过程中应该注意的几个问题。

对于一个较为完善的输入系统来说，当用户键入非法输入时，该系统应该向用户提示“非法输入”等错误信息，并提示用户重新输入。另外，当用户用 TAB 键移动焦点到文本框中，如果文本框中有缺省值，该缺省值应该以高亮度字符显示，这样用户输入的字符就可以直接覆盖缺省值而不要求用户先清除该缺省设置，下面的例子就讲上面问题的实现过程。

用户可以在窗体上添加三个文本框分别用来显示年、月、日等信息，并且添加一个命令按钮用来结束程序运行，各控件的属性设置如下表，建立的初始界面为图8.15。

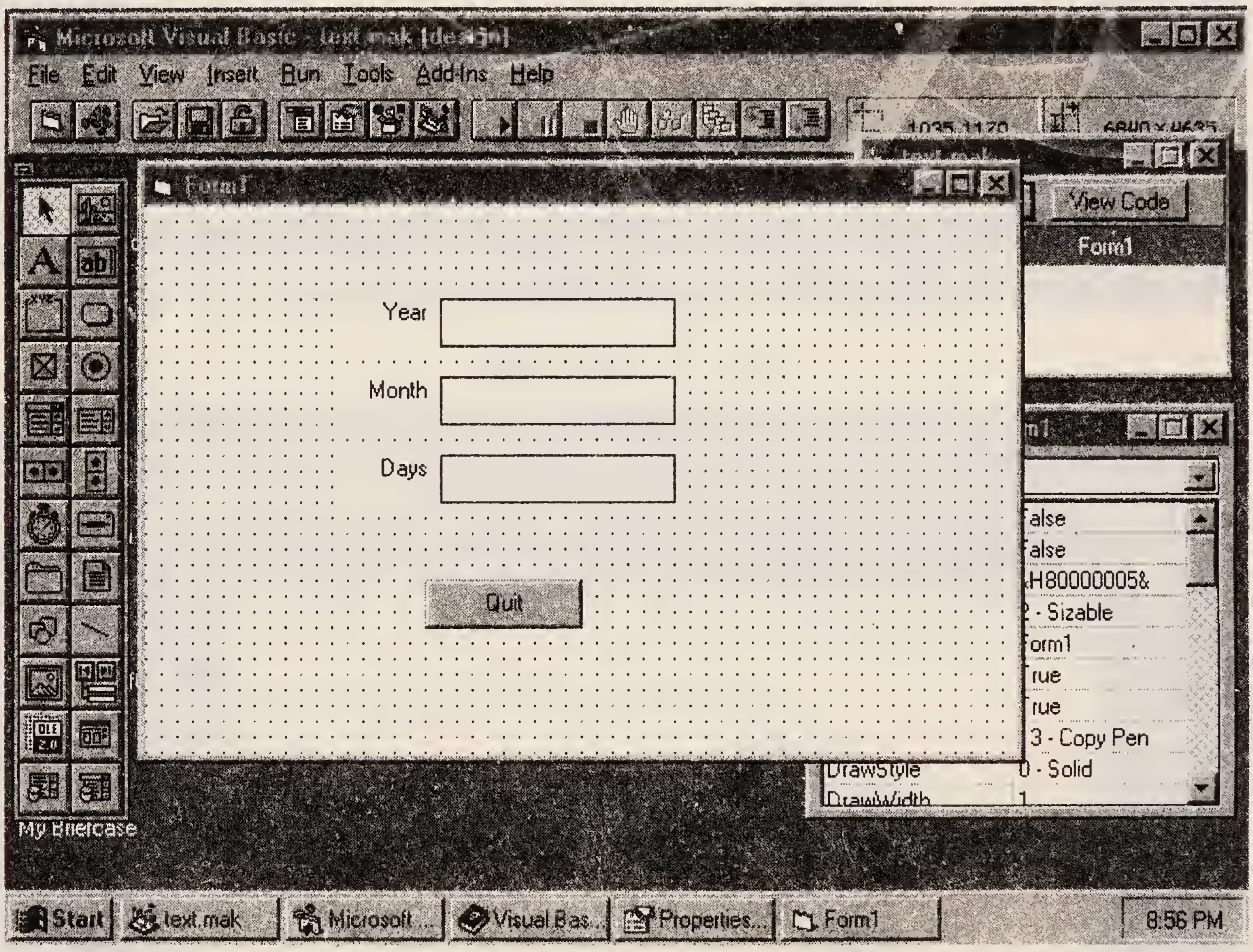

Microsoft Visual Basic - text.mak [design]
File
Edit
View
Insert
Run
Tools
Add-Ins
Help
Form1
View Code
Form1
Year
Month
Days
Quit
False
False
H80000005&
2 - Sizable
Form1
True
True
3 - Copy Pen
DrawStyle
0 - Solid
DrawWidth
1
My Briefcase
Start
text.mak
Microsoft ...
Visual Bas...
Properties...
Form1
8:56 PM

图8.15 初始化界面

控件	Name	Text	Caption
文本框1	Text1	空白	没有
文本框2	Text2	空白	没有
文本框3	Text3	空白	没有
命令按钮	Command1	没有	Quit
标签1	Label1	没有	Year
标签2	Label2	没有	Month
标签3	Label3	没有	Days

为检查文本框中输入的合法性，用户可以在相应的事件过程中编写以下代码：

```
Sub Form _ load()
  Dim Today
  Today=Now
  Text1. Text = Year(today)
  Text2. Text = Month(today)
  Text3. Text = Day(today)
End Sub
```

以上代码用 Now 函数取得当今时间并把它赋给变量 Today。然后分别用 Year、Month、Day 函数从当今时间中取出年、月、日作为文本框的缺省设置，为了使当焦点移动到文本框中时缺省值被以高亮度字符显示，必须编写以下代码：

```
Sub Text1 _ GotFocus()
  Text1. SelStart=0
  Text1. SelLength=Len(Text1. Text)
End Sub
Sub Text2 _ GotFocus()
  Text2. SelStart=0
  Text2. SelLength=Len(Text2. Text)
End Sub
Sub Text3 _ GotFocus()
  Text3. SelStart=0
  Text3. SelLength=Len(Text3. Text)
End Sub
```

以上代码用文本框的 SelStart 和 SelLength 属性选择了文本框中的文本。因为年、月、日都有一定界限的，为了对于非法输入程序有所反应，可以编写以下代码：

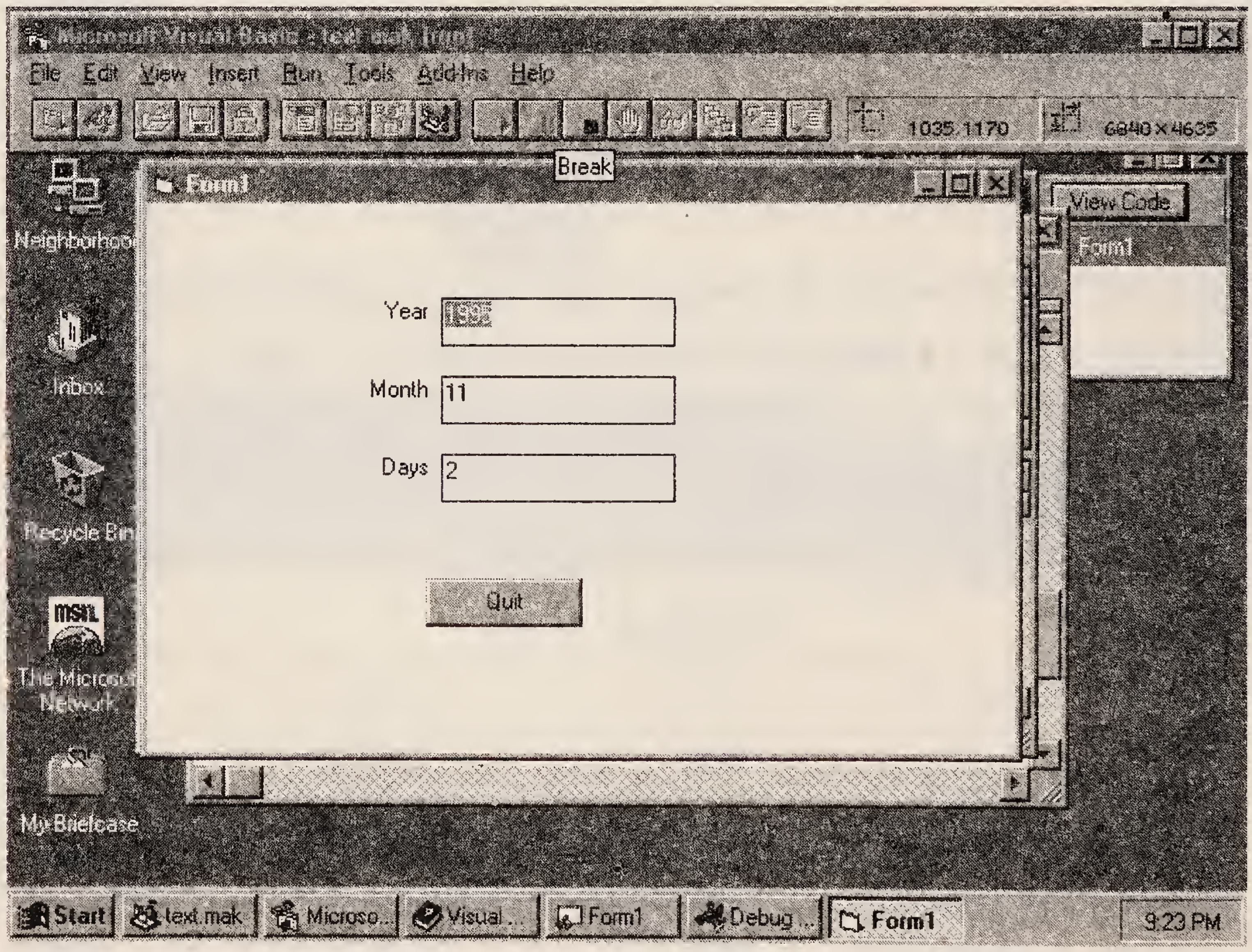
```
Sub Text1 _ Change()
    Dim Year As integer
 If Year = Val(Text1. Text)<0 Then
    MsgBox " illeagal input"
    Exit Sub
  End If
End Sub
Sub Text2 _ Change()
    Dim month As integer
 If month = Val (Text2. Text) < 0 or Val(Text2. Text) >12 Then
    MsgBox "illeagal input"
```

```
        Exit Sub
      End If
    End Sub
    Sub Text3 _ Change()
        Dim Days As integer
     If days = Val (Text3. Text) < 0 or Val(Text3. Text) > 31 Then
        Msg Box "illeagal input"
        Exit Sub
     End If
    End Sub
```

为了能够方便地结束程序，还应编写以下代码：

```
    Sub Command1 _ Click()
             End
    End Sub
```

当用户单击工具条上的 Start 按钮运行程序时，屏幕上的文本框显示当今时间，并且年份的缺省值以高亮度字符显示，见图8.16。当用户键入非法值，例如字母或超过年、月、日范围的值，屏幕上会弹出一消息框提示用户输入错误，见图8.17。用户单击 Quit 按钮可结束程序。



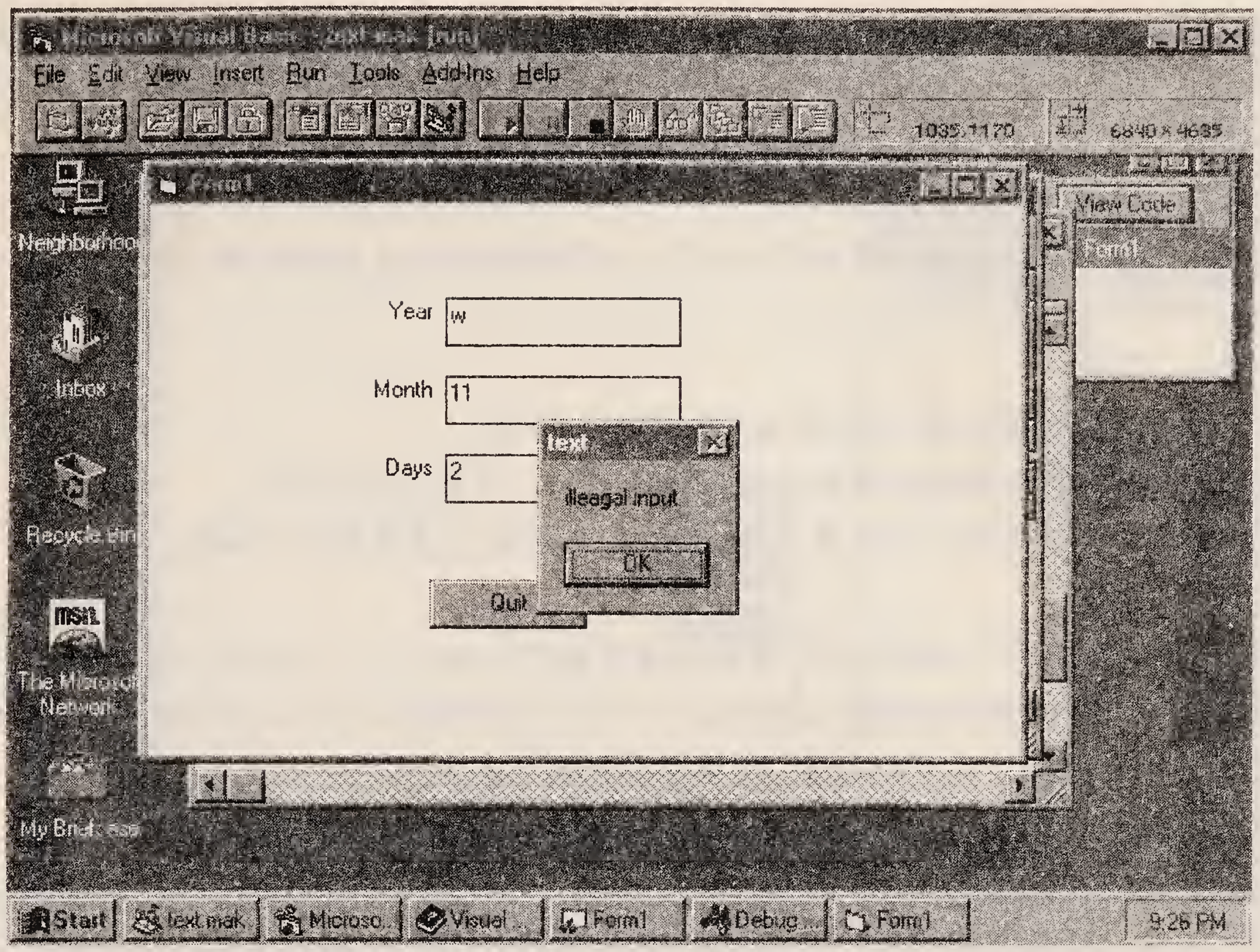
Year [illegible]

Month 11

Days 2

Microsoft Visual Basic - [text.mak [run]]
File
Edit
View
Insert
Run
Tools
Add-Ins
Help
1035,1170
6840 x 4635
Break
Form1
View Code
Form1
Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Quit
Start
text.mak
Microso...
Visual...
Form1
Debug...
Form1
9:23 PM

图8.16　显示缺省设置


File Edit View Insert Run Tools Add-Ins Help
1035,1170
6840 x 4695
Form1
View Code
Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
text
OK
Quit
Start
text.mak
Microso...
Visual...
Form1
Debug...
Form1
9:26 PM

Year w

Month 11

Days 2

illeagal input

图8.17 提示非法输入

8.3.4 列表框的建立

列表框在对话框中经常看到，它显示可供用户选择的选项，用户可单击选项来选择它。如果用户想在窗体中放置列表框可使用工具箱中的列表框工具。在列表框的建立过程中要经常使用 AddItem、RemoveItem、Clear 三种方法，它们可在运行时修改列表框的内容。

(1) AddItem 方法

该方法在运行时向列表框或组合框中添加新项目，或往网格控件中增加新的一行。语法表达式如下：

```
Control. Additem item[,index]
```

语法表达式中 Control 表示列表框、组合框或网格等控件，item 表示加到控件中的字符串表达式，只有在网格控件中使用制表字符(字符代码为09)来分隔要插入到新增加行的每一列中的多个字符串。index(整数)表示新增加的项或行在控件中的位置，例如，对列表框或组合框中第一项或网格控件的第一行 index=0。如果用户为 index 指定了一个有效值，item 将放在控件中的那个位置上，若省略 index，item 就放在某个排序位置上(当 Sorted 属性为 True 时)，或放在列表的末尾(当 Sorted 属性为 False 时)。

(2) RemoveItem 方法

该方法能够从列表框或组合框中删除一项，或者从网格中移去一行，语法表达式为：

```
Control. RemoveItem index
```

其中 index 代表要删除的项或行的位置，例如，用户要删除列表框或组合框的第一项时，index＝0。如果一个列表框或一个组合框被约束给数据控件时，RemoveItem 方法不再适用。

(3) Clear 方法

该方法删除列表框或组合框中所有项目，或者清除操作环境剪贴板上所有内容，语法的表达式为：

Object.Clear

其中 Object 代表组合框、列表框或剪贴板(Clipboard)。

如果列表框或组合框被约束给数据控件时，Clear 方法不再适用。

列表框控件除了使用上述方法以外，在建立对话框中还常用以下属性。

(1) Text 属性

该属性决定列表框中被选的项，返回值通常等于表达式 List(ListIndex)的值，设计时不能用，运行时只读。该属性也适合组合框，对于下拉式列表框，它也是决定列表中被选中的项；对于简单组合框和下拉式组合框，该属性决定包含在编辑区中的文本。

(2) List 属性

该属性确定列表框的列表部分的项，列表是一个字符串数组，在串数组中每个单元是一个列表项。用这个属性可以访问列表框的列表项，例如，可以用 List(0)、List(1)、List(2)分别访问列表框的第一、二、三项。该属性也适用于后面讲到的组合框。

(3) ListIndex 属性

该属性确定列表框当前被选项的索引。如果该属性被设置成－1指示当前没有选中项，则表达式 List(List Index)返回当前被选中的项。该属性也适用后面讲到的组合框。不过被设置为－1时表示用户已经输入新文本到组合框的文本框部分。

(4) ListCount 属性

该属性指定在列表框中列表部分中项的数目。

为能够对建立列表框有进一步的认识，下面我们将举一个例子。该例用来在列表框中加入或删除项目，具体步骤如下。

(1) 单击 File 菜单的 New Project 命令，创建一个新工程，建立如图8.18所示的初始化界面，其中属性设置如下表。

控件	Name	Caption	Text
窗体	Form1	example of list box	没有
标签	Label 1	input	没有
列表框	List 1	没有	没有
文本框	textnum	没有	没有
命令按钮1	Cmadditem	additem	没有
命令按钮2	Cmremoveitem	removeitem	没有
命令按钮3	CmClose	Close	没有

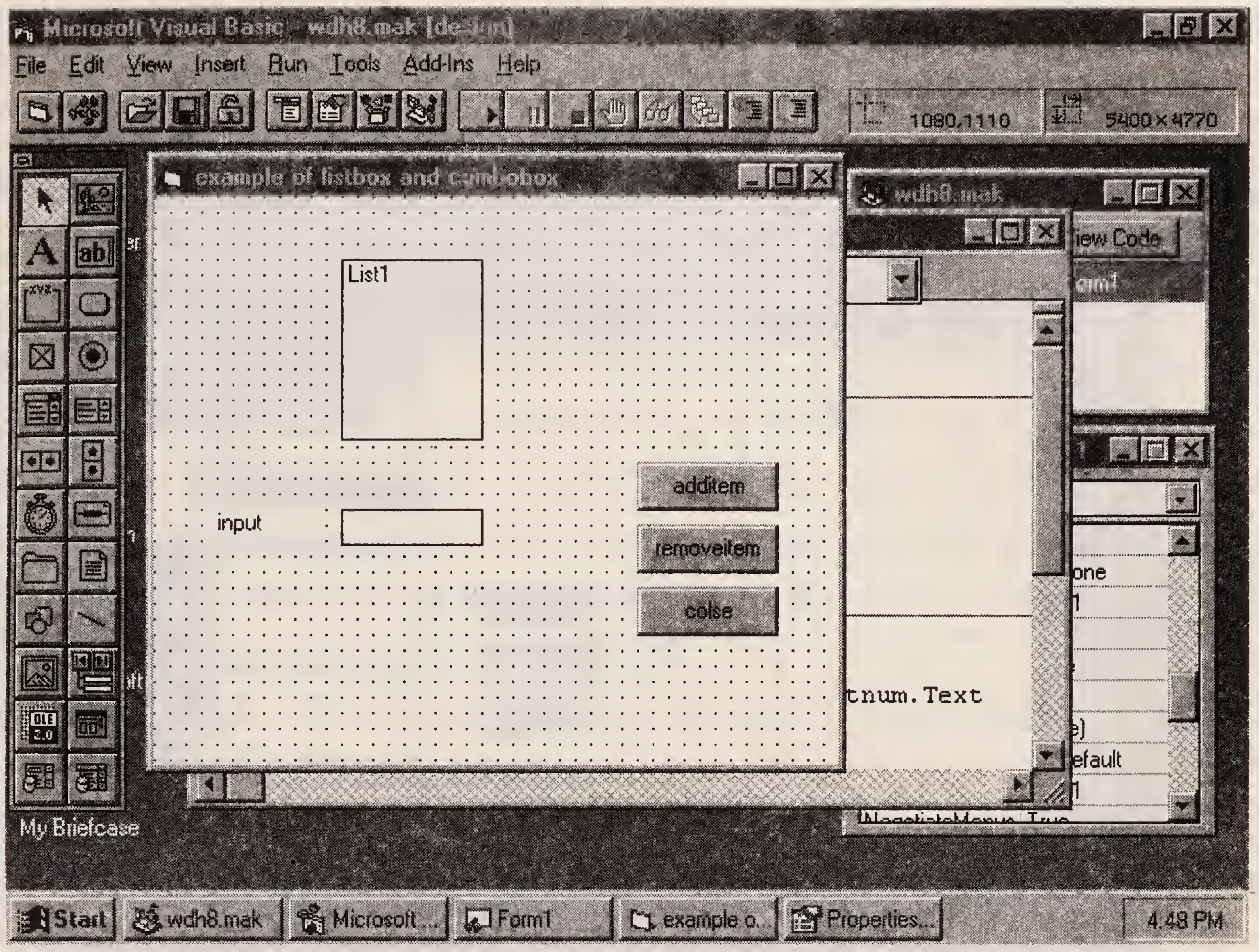

Microsoft Visual Basic - wdh8.mak [design]
File Edit View Insert Run Tools Add-Ins Help
1080,1110
5400 x 4770
example of listbox and cumbobox
List1
input
additem
removeitem
colse
wdh8.mak
View Code
tnum.Text
My Briefcase
Start
wdh8.mak
Microsoft...
Form1
example o...
Properties...
4:48 PM

图8.18 初始化界面

（2）编写代码

```
Sub Form _ Load()
  List1. Additem "wdh"
  List1. Additem "Zb"
End Sub
```

该事件在窗体加载时向列表框中添加项目。

```
Sub Comadditem _ Click()
 If textnum<>" " Then List1. Additem textnum. text
    textnum. text=" "
    textnum. SetFocus
End Sub
```

该事件在单击 additem 按钮时发生，当文本框中有内容时就加到列表框中，第二条语句清除文本框，第三条语句把焦点放在 Textnum 后面，因而用户在加入前一条项目后可立即开始加另一个新项目。

```
Sub Cmremoveitem _ Click()
 If List1. ListIndex<0 Then Exit Sub
    List1. Removeitem List1. Listindex
End Sub
```

该事件在单击 Removeitem 命令按钮时发生。如果列表框中没有项目，则从过程中退出，否则删除所选项目，同时用 Listindex 值来指定哪一个项目将被删除。

```
Sub Cmclose _ Click()
   End
End Sub
```

该事件在单击 Close 命令按钮时发生，用以退出程序。

（3）按 F5键运行程序，可以看到列表框中显示出两项“wdh”和“Zb”。

（4）单击“Zb”项目，然后单击 removeitem 按钮就可删除“Zb”项目，见图8.19。

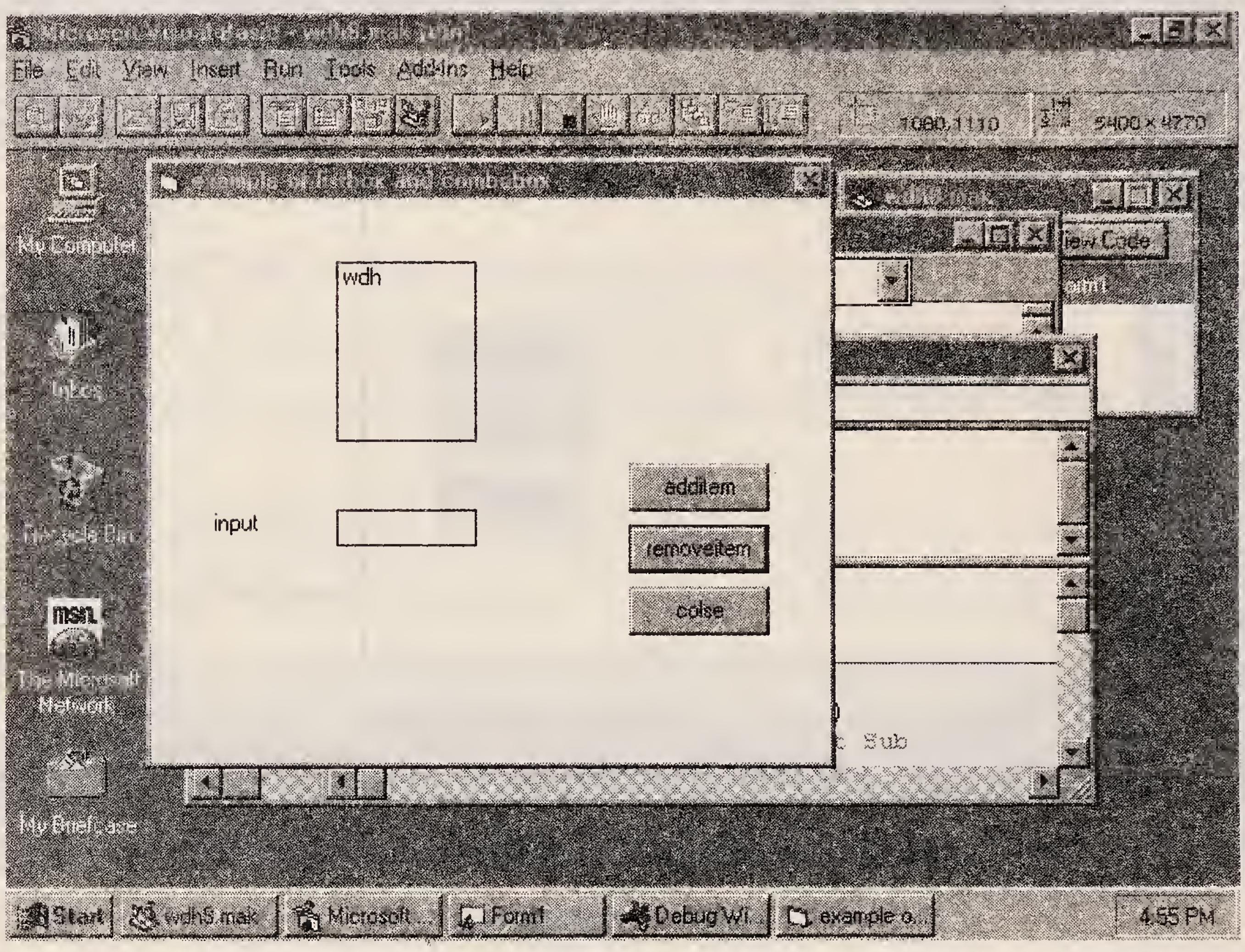

File Edit View Insert Run Tools Add-Ins Help
1080,1110
5400 x 4770
My Computer
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
wdh
input
additem
removeitem
colse
Sub
Start
wdh8.mak
Microsoft...
Form1
Debug Wi...
example o...
4:55 PM

图8.19 删除一项

（5）在文本框中单击并输入“Visual Basic 4.0”，然后单击 additem 按钮，文本框内容被加到列表框中，见图8.20。

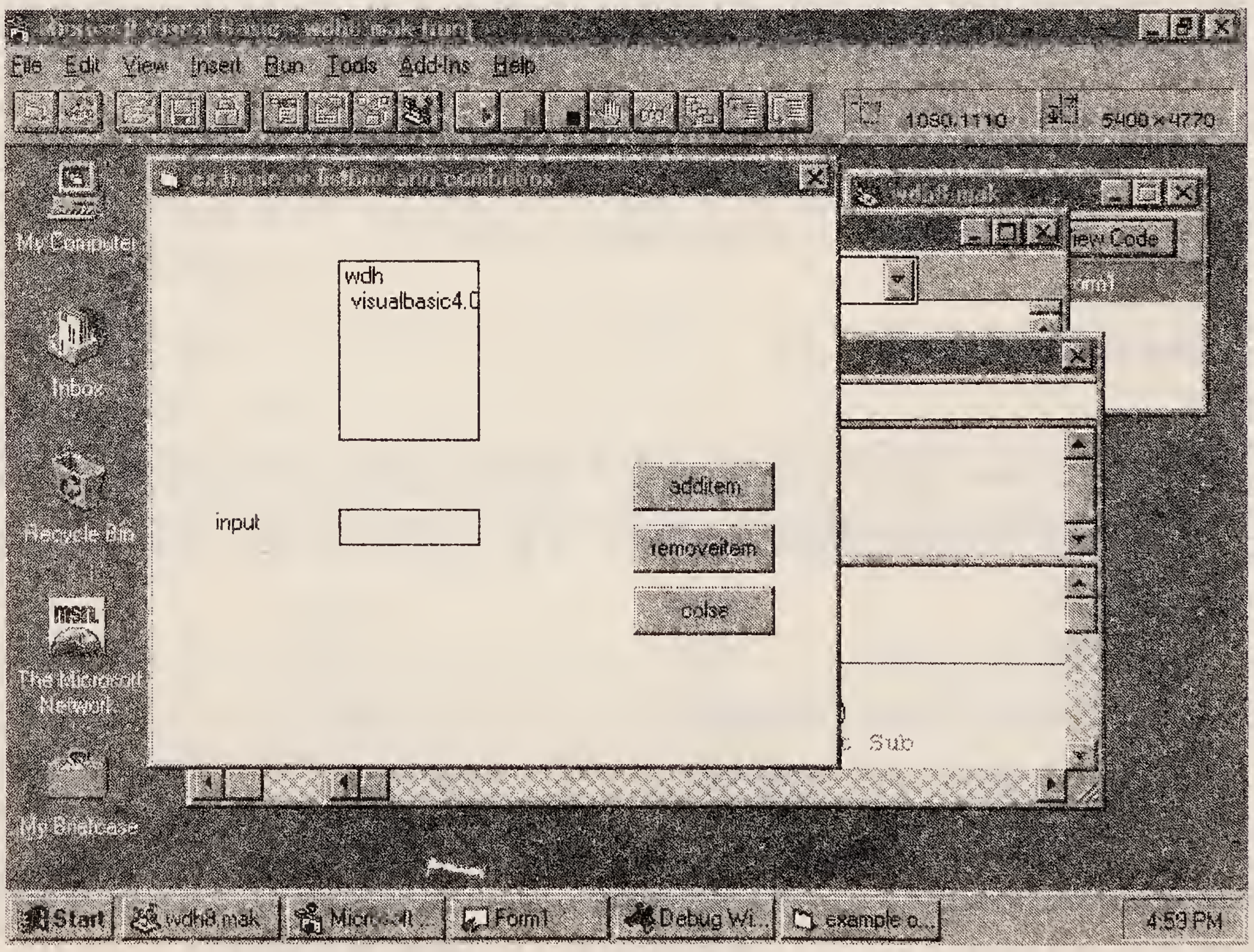

File Edit View Insert Run Tools Add-Ins Help
1080,1110
5400 x 4770
My Computer
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
wdh
visualbasic4.
input
additem
removeitem
colse
Sub
Start
wdh8.mak
Microsoft...
Form1
Debug Wi...
example o...
4:59 PM

图8.20 加入一项

（6）单击 Close 结束程序运行。

8.3.5 组合框的建立

在“窗体与工具”一章中用户已知道所谓组合框就是组合文本框和列表框而成的控件，组合框有三种不同的类型，这由组合框的 Style 属性确定，属性 Style 被设置为0（缺省）时，组合框称为下拉式组合框，用户可以在其文本框中直接键入文本或单击右边的箭头打开列表供用户选择。Style 被设置为1时，组合框称为简单组合框，它列出所有的项目供用户选择，用户也可以在文本框中输入项目。Style 被设置为2时，组合框称为下拉式列表框，它允许用户单击右边的箭头打开列表供用户选择，不允许用户直接键入自己的项目。

与组合框相关的事件除 ChangeClick、GotFocus、LostFocus 等等以外，还常用到 DropDown 事件，当用户单击组合框下箭头时发生该事件，该事件实质上对应于下箭头的“单击”事件，但用户必须注意该事件不适用于简单组合框。

在组合框中常用像 Additem 和 RemoveItem 等方法，它们的用法也一样，在此不多述。下面举一个简单的例子说明组合框的建立。

下面的例子唯一目的是检查组合框中项目的排序状态，在设计时组合框的 Sorted 属性为 True，此时按顺序排列。当用户用指定索引号的 Additem 方法向组合框中加入一项时，组合框中字母还按顺序排列吗？下面我们看程序的执行结果。

首先建立如图8.21所示的初始化界面，各控件的属性设置如下。

窗体	Caption 为 Combo Box test
组合框	Name 为 Combo1，Style 为2，Sorted 为 True
命令按钮1	Name 为 CmAdd，Caption 为 Add
命令按钮2	Name 为 CmRemove，Caption 为 Remove
文本框1	Name 为 Chosen，Text 为空白
文本框2	Name 为 SortState，Text 为空白
文本框3	Name 为 Message，Text 为空白
标签1	Caption 为 List item chosen
标签2	Caption 为 Sort State of List
标签3	Caption 为 Message of List

接下来的工作是编写代码。打开代码窗口，在相应事件过程中键入如下代码：

```
Sub Form _ Load()
   Combol. Additem "a"
   Combol. Additem "g"
   Combol. Additem "h"
   Combol. Additem "i"
   Combol. Additem "k"
End Sub
Sub CmAdd _ Click()
   Combol. Additem "b",3
   SortState. Text=Str $ (Combol. Sorted)
End Sub
```

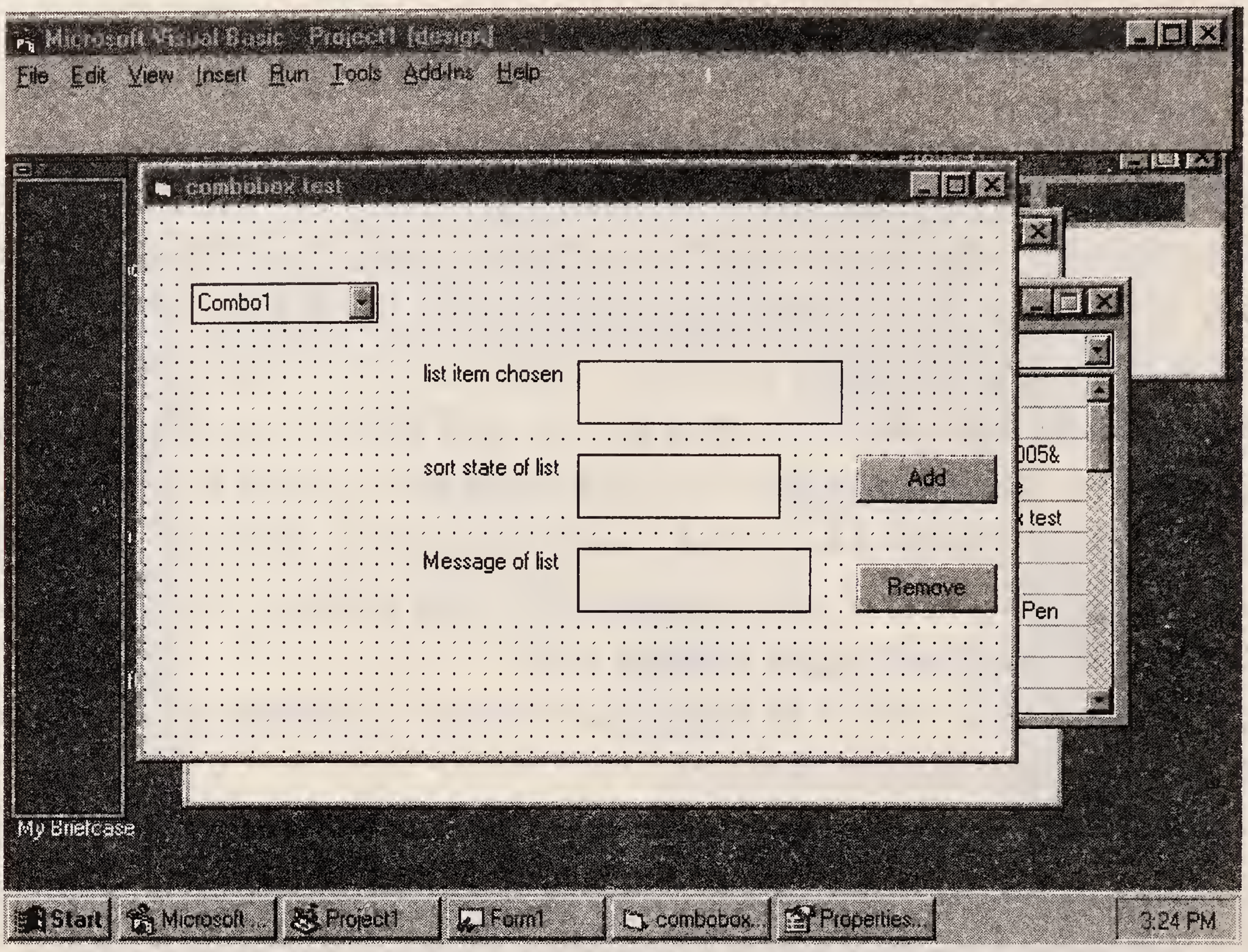
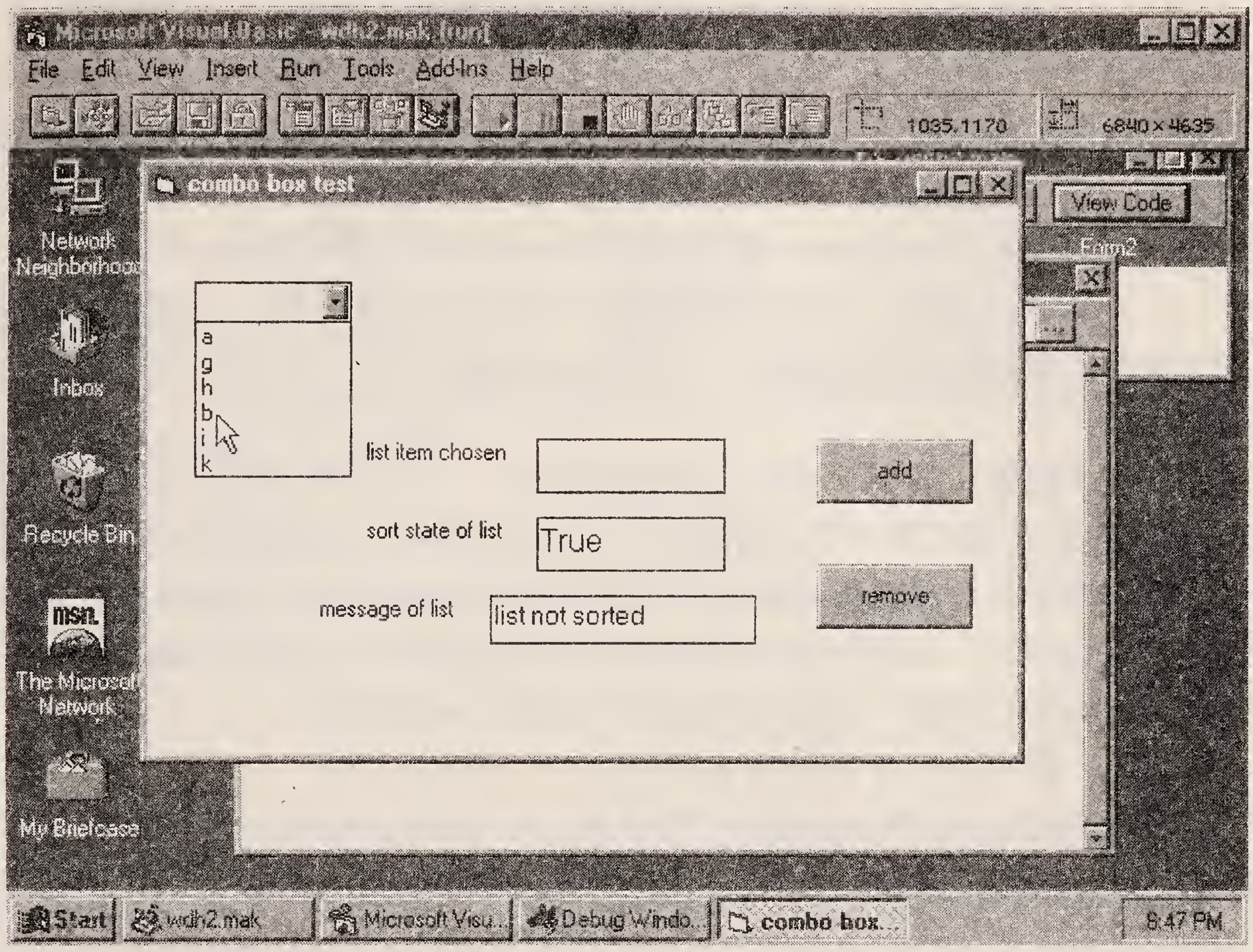

Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
combobox test
Combo1
list item chosen
sort state of list
Message of list
Add
Remove
My Briefcase
Start
Microsoft...
Project1
Form1
combobox...
Properties...
3:24 PM

图8.21　建初始化界面

```
Sub CmRemove _ Click()
   Combol. Removeitem 3
   SortState. Text=Str $ (Combol. Sorted)
End Sub
Sub Combol _ Click()
   Chose. Text=Str $ (Combol. ListIndex)
   SortState. Text=Str $ (Combol. Sorted)
End Sub
Sub Combol _ DropDown()
  For number=1 To Combol. ListCount-1
     Str1 $ =Left $ (Combol. List(number),1)
     Str2 $ =Left $ (Combol. List(number-1),1)
  If Asc(Str1 $ )<Asc(Str2 $ )Then
     Message. Text=″List not Sorted″
     Exit Sub
     End If
     Message. Text=″ ″
End Sub.
```

按 F5键运行程序，单击组合框的箭头打开列表可以看到字母按顺序排列，当用户单击 Add 按钮向列表加入一项“b”时，此时再单击组合框箭头，用户可以看到“b”被加到列表项中，这时 Message of List 的文字框显示“List not Sorted”表明没有按序排列，Sort State of List 的文本框显示 False，见图8.22。这说明用指定索引的 Additem 方法加入一项时，删除改变 Sorted 属性。当用户单击 Remove 按钮删除 b，再单击组合框的箭头时，Message of List 文

本框没有显示信息。Sorted 属性为 True 说明字母按顺序排列，见图8.23。


Microsoft Visual Basic - wdh2.mak [run]
File Edit View Insert Run Tools Add-Ins Help
1035,1170
6840 x 4635
View Code
Form2
Network Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
add
remove
Start
wdh2.mak
Microsoft Visu...
Debug Windo...
combo box...
6:47 PM

combo box test

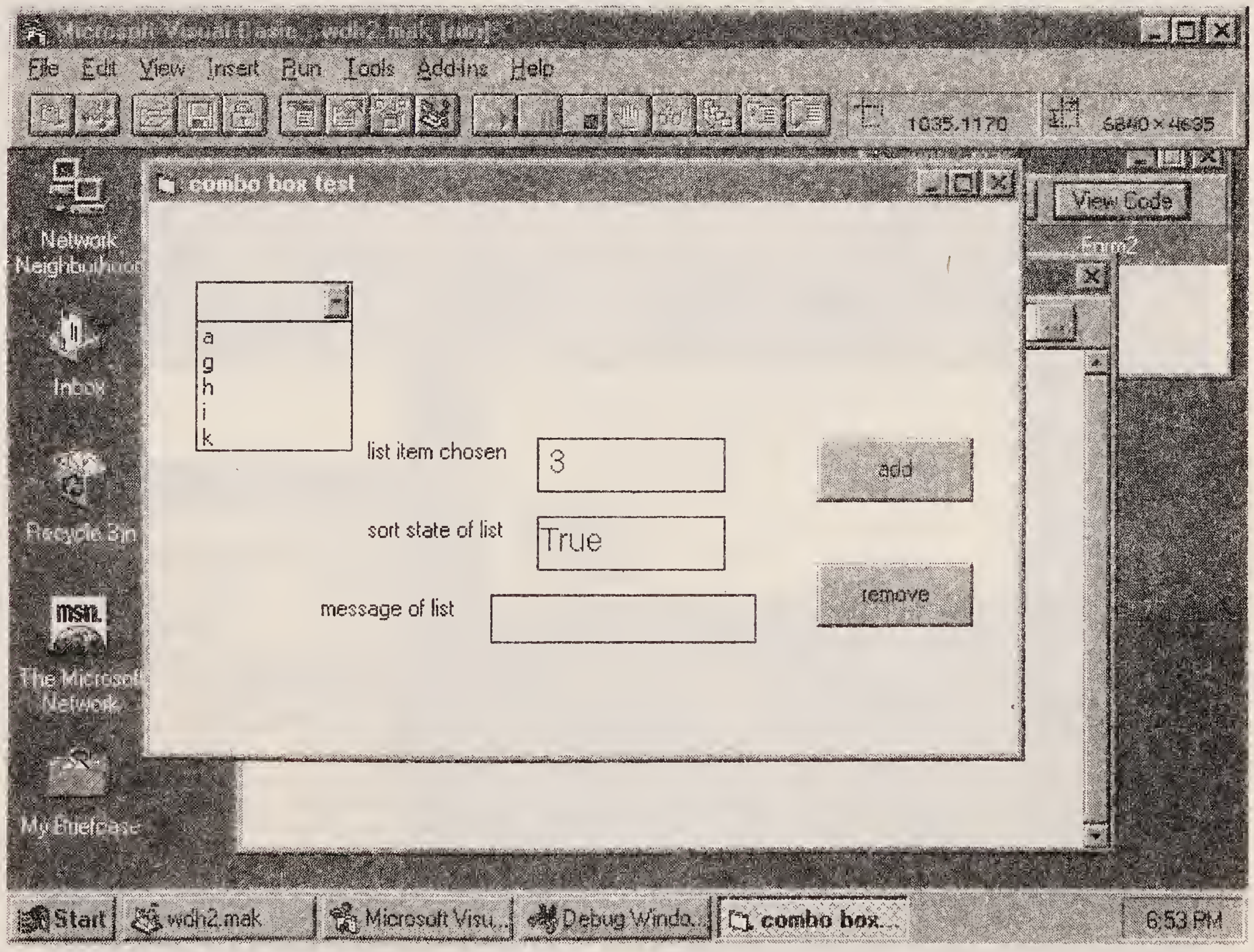
a
g
h
b
i
k

list item chosen

sort state of list: True

message of list: list not sorted

图8.22 不按顺序排列


Microsoft Visual Basic - wdh2.mak [run]
File Edit View Insert Run Tools Add-Ins Help
1035,1170
6840 x 4635
View Code
Form2
Network Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
add
remove
Start
wdh2.mak
Microsoft Visu...
Debug Windo...
combo box...
6:53 PM

combo box test

a
g
h
i
k

list item chosen: 3

sort state of list: True

message of list

图8.23 按顺序排列

8.4　定制对话框

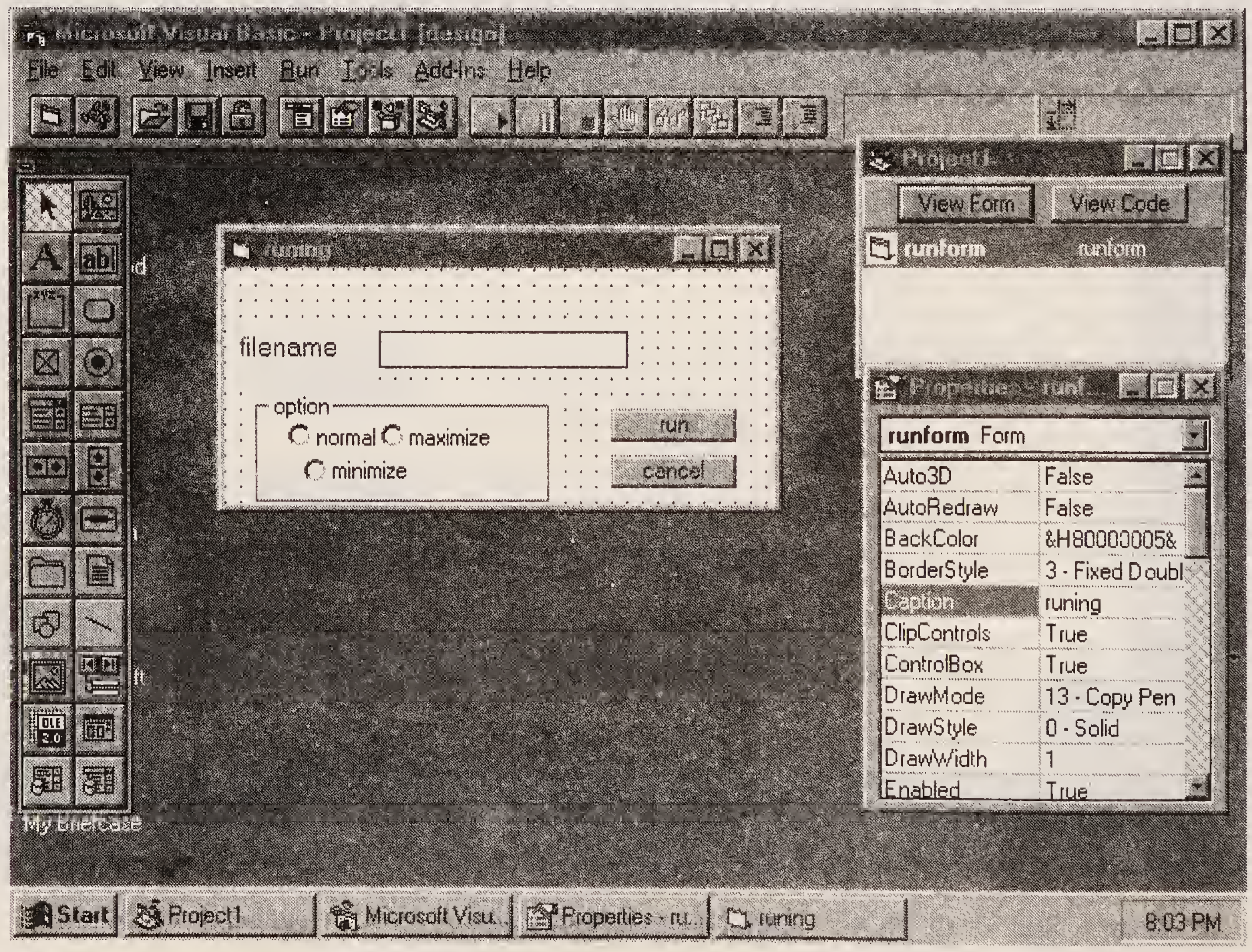
上节举例说明了各种控件的建立，实际上它们都是一个对话框，因为对话框就是一个含有控件的窗体。本节不打算讲解一个更为复杂的对话框的建立过程，例如像 Open 或 Save as 等对话框的建立，因为 Visual Basic 提供的公用对话框控件能够使用户方便地建立诸如 Open、Save as、Print 等对话框，有关的内容将在下节讲解。本节还是以一个例子为线索，讲解一个常用但又不太复杂的对话框的建立，主要目的是使用户对一个定制对话框的建立有一个较全面的了解。

或许用户需要经常运行一个程序，如果能够建立一个运行程序的对话框将使用户运行一个程序更为方便，就像在第一节讲到的那样，大多数情况下没有理由对对话框重设大小，所以最好把运行对话框设置成双边固定框。对于运行对话框也没有必要有最大化按钮和最小化按钮，但是对话框左上端的控制菜单按钮是有必要保留的，这是因为对没有鼠标的用户将很方便地关闭运行对话框，下面是建立运行对话框的一般步骤。

(1) 单击 File 菜单的 New Project 命令，建立一个新工程文件。

(2) 设置对话框的边界风格，用户可以根据运行对话框中控件的多少来安排对话框的大小，本例对话框的边界风格设置如下：Name 属性为 RunForm，Caption 属性为 Running，Height 属性为2295，Width 属性为3885，ControlBox 设置为 True，BorderStyle 为3-FixedD，MaxButton 和 MinButton 为 False。

(3) 建立初始化界面，在窗体上加载以下控件并设置其属性为下表所示，建立如图8.24所示的初始化界面。


Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
runing
filename
option
normal
maximize
minimize
run
cancel
Project1
View Form
View Code
runform
runform
Properties - runf
My Briefcase
Start
Project1
Microsoft Visu...
Properties - ru...
runing
8:03 PM

runform Form

Auto3D	False
AutoRedraw	False
BackColor	&H80000005&
BorderStyle	3 - Fixed Doubl
Caption	runing
ClipControls	True
ControlBox	True
DrawMode	13 - Copy Pen
DrawStyle	0 - Solid
DrawWidth	1
Enabled	True

图8.24　初始化界面

控件	Name 属性	Caption 属性	Text 属性
文本框	RText	没有	空白
标签	Label1	filename	没有
框架	Frame1	Option	没有
单选钮1	Noption	Normal	没有
单选钮2	Maxoption	Maximize	没有
单选钮3	Minoption	Minimize	没有
命令按钮1	CmRun	Run	没有

（4）编写代码，双击窗体的空白处打开代码窗口，在对象列表框和过程列表框中选择相应的对象及事件过程编写以下代码：

```
Sub Cm Run _ Click()
    On Error GoTo errorhandler
    If Noption Then retval=Shell(RText. Text,1)
    If Maxoption Then retval=Shell(RText. Text,2)
    If Minoption Then retval=Shell(RText. Text,3)
    Exit Sub
    errorhandler:
    MsgBox "Cann't run that file"
    Resume Next
End Sub
```

在上述代码中用了一个 Shell 函数，Shell 函数可以用来运行一个程序，它的用法如下：

```
Shell(Command String,WindowStyle)
```

Command String 参数代表所要执行的程序名，不在当前目录的情况下，应该包括完整的路径名。如果程序名没有包括.BAT，.COM，.EXE 扩展名，则扩展名假定为.EXE。Window Style 参数决定程序所在窗口的风格，下表列出了 WindowStyle 值及相对应的风格。

值	窗口风格
1,5,9	正常方式并拥有焦点
2	（缺省）最小化方式并拥有焦点
3	最大化方式并拥有焦点
4,8	正常方式并无焦点
6,7	最小化方式并无焦点

（5）运行程序

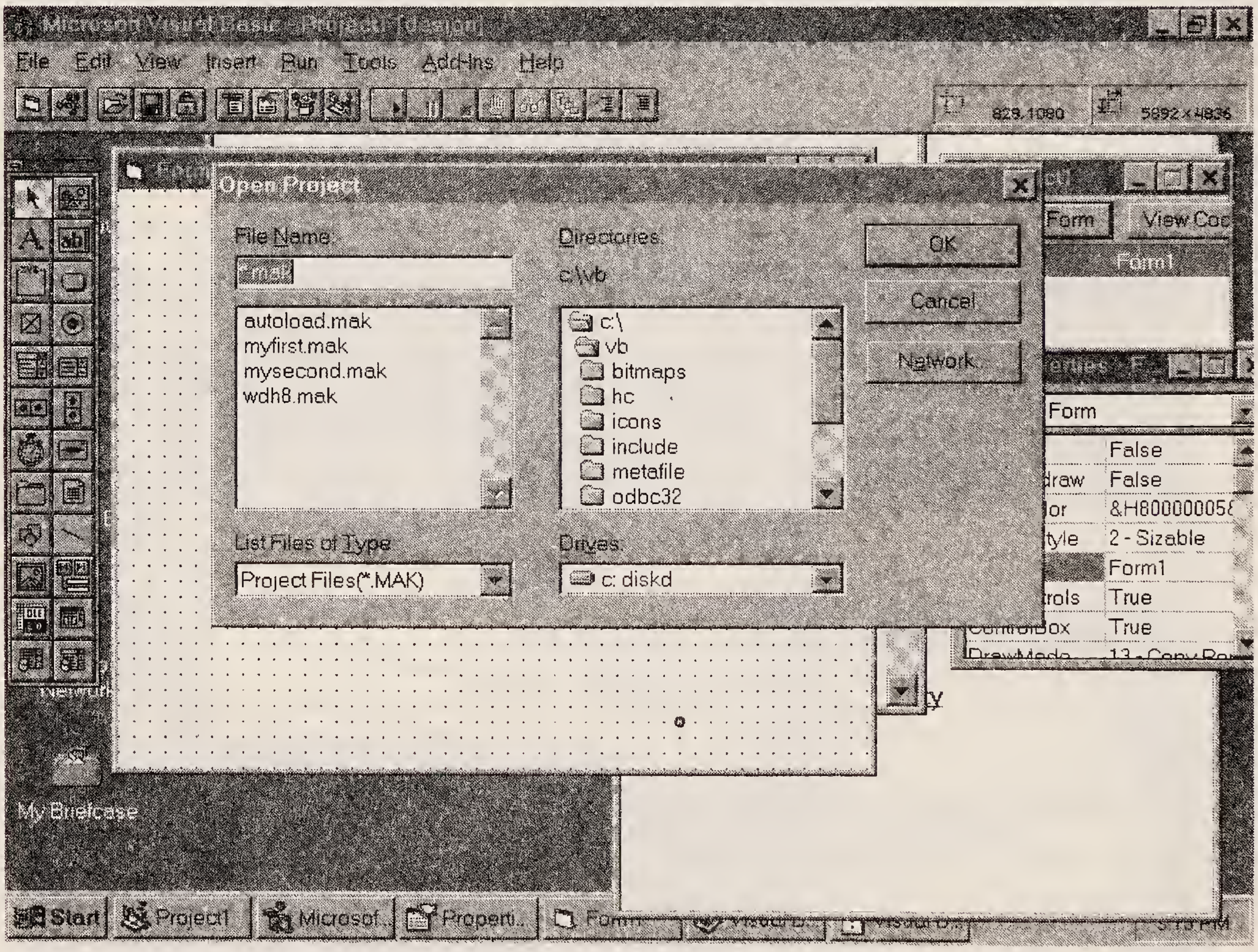
用户按 F5键运行程序后在对话框中键入程序名即可运行一个程序。例如，键入 C:\Windows95\Calc.EXE 就可运行计算器程序。

8.5　公用对话框

如果需要简单的消息框或输入框，函数 MsgBox 和 InputBox 是相当有用的，但是在很多情况下，用户程序需要相当复杂的对话框，例如像图8.25一样的 Open 对话框，对于这样一个对话框，如果让用户来编写一定很麻烦，更会使初学者望而怯步。幸运的是，Visual Basic 4.0为用户提供了公用标准的 Windows 对话框：Open、Save As、Color、Font 和 Print，另外公

用对话框还能够启动帮助系统。要使用公用对话框，必须在项目窗口加入 COMDLG16.OCX 或 COMDLG32.OCX，用户可利用 File 菜单中的 Add File 命令把 CMDIALOG 文件加到项目窗口中。如果用户想在新项目文件中自动包括 COMDLG16.OCX 或 CMDLG32.OCX，可把它放在 AUTOLOAD.MAK 文件中。在应用程序中使用公用对话框，需要把公用对话框控件添加到窗体并设置它的属性。对话框的类型由 Action 属性决定，运行时 Action 属性设置有效值时显示对话框；在设计时，公用对话框在窗体中以图标形式显示，这时用户不能够调整图标的尺寸。使用公用对话框代替许多定制对话框有许多优点，它能够保证用户的应用程序和其它 Windows 程序有一致接口，它不需要用户增加任何程序代码就能够在用户程序中增加功能。公用对话框类型与 Action 属性的对应关系如下表。

Action 属性	对话框类型
0	无任何动作
1	Open 对话框
2	Save As 对话框
3	Color 对话框
4	Font 对话
5	Print 对话框
6	调用 help 文件


Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
823,1080
5892 x 4836
Open Project
OK
Cancel
Network...
autoload.mak
myfirst.mak
mysecond.mak
wdh8.mak
c:\
vb
bitmaps
hc
icons
include
metafile
odbc32
View Form
View Code
Form1
Form
My Briefcase
Start
Project1
Microsof...
Properti...
Form1
5:13 PM

File Name: *.mak

Directories: c:\vb

List Files of Type: Project Files(*.MAK)

Drives: c: diskd

False
False
&H8000000S
2 - Sizable
Form1
True
True

图8.25 Open project 对话框

8.5.1 Open 对话框

1. 属性

(1) Default 属性

该属性设置对话框缺省时的文件扩展名，当用户存储或打开没有扩展名的文件时，由此属性指定的扩展名自动加到文件名上。

(2) DialogTitle 属性

该属性用来设置对话框中标题条上显示的字，Open 对话框的缺省标题是 Open。

(3) FileName 属性

该属性用来设置或返回打开或存储文件的路径和名称，当用户选择一个文件并单击对话框的 OK 按钮时，就给 FileName 属性设置了值，然后可用于打开或存储所选中的文件。

(4) Filter 属性

该属性用来显示在对话框中的 Type 列表框中的过滤器，可使用此属性给用户提供可供选择的文件类型，Filter 属性必须是一个具有一对或多对组成串的正文串，组成串由描述符和通配符组成，并由管道符“|”隔开，多对也必须由管道符隔开。语法格式为：

```
[Form. ]CMDialog. Filter=(description|filter1|description|filter)。
```

例如：

```
Form1. CMDialog. Filter=(*. DOC)|*. DOC|Textfiles|*. TXT|All Files|*. *
```

(5) FilterIndex 属性

该属性指定缺省的过滤器，设置值为一整数。

(6) Flags 属性

为打开或存储对话框设置选择项，相应的符号常量被列在 Visual Basic 中 Object Browser 下的对象库中。

(7) InitDir 属性

确定初始化文件目录。

(8) MaxFileSize 属性

该属性用于设定文件名(含路径)的最大长度。

(9) CancelError 属性

该属性确定当用户选择按钮时是否产生错误。如果设置为 True，则产生错误；若设置为 False(缺省)，则不产生错误。

(10) HelpCommand 属性

该属性指定所需 Help 的类型，它的特性值在 Visual Basic 的 CONSTANT. TXT 文件中有定义。例如 HELP _ KEY 是显示某一特定字的 Help。

(11) HelpContext 属性

该属性确定所需 Help 的 ID 内容，它与 HelpCommand 属性(设置 HelpCommand=HELP _ CONTEXT)一起使用，指示显示的 Help 主题。

(12) HelpFile 和 HelpKey

它们分别指定 Help 应用程序的 Help 文件名和 Help 主题能够识别的关键字。

2. 建立 Open 对话框

当把1赋给公用对话框的 Action 属性时，就可以启动 Open 对话框，用户可以根据上面的属性设置来控制 Open 对话框的外观，用户必须注意 Open 对话框并不真正打开文件，它只允许用户选择文件名，如果用户在选一文件名后，想打开该文件必须调用打开文件的子过程，以下程序段，在用户单击菜单的 Open 命令时将显示一对话框，见图8.26。

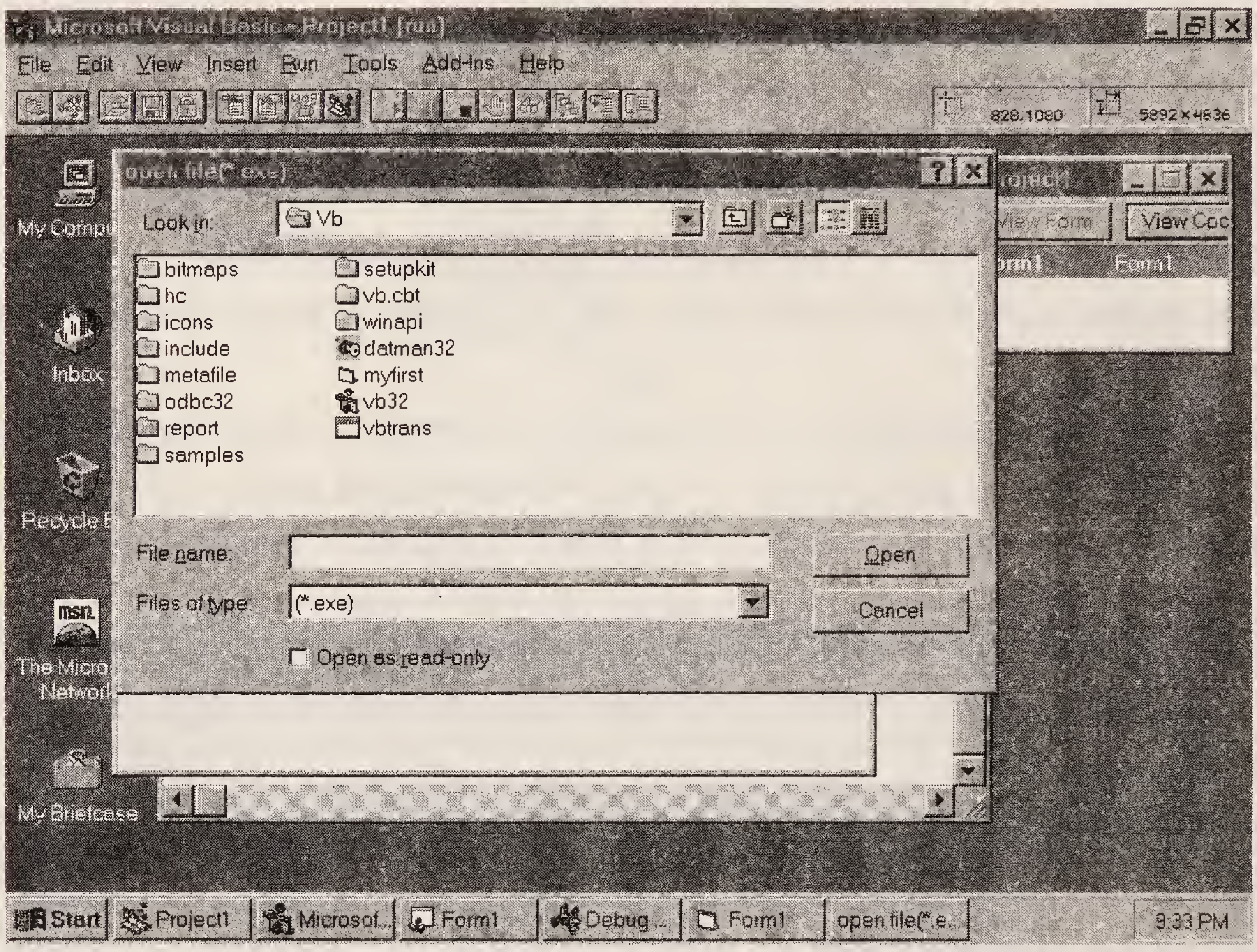

Microsoft Visual Basic - Project1 [run]
File Edit View Insert Run Tools Add-Ins Help
828,1080
5892×4636
open file(*.exe)
Look in:
Vb
bitmaps
hc
icons
include
metafile
odbc32
report
samples
setupkit
vb.cbt
winapi
datman32
myfirst
vb32
vbtrans
File name:
Files of type:
(*.exe)
Open
Cancel
Open as read-only
View Form
View Code
Form1
My Computer
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Start
Project1
Microsof...
Form1
Debug ...
Form1
open file(*.e...
3:33 PM

图8.26 Open 对话框

```
Sub menu open _ Click()
    CMDialog1.FileName= " "
    CMDialog1.FlagS=VbOFN FILEMUSTEXIST
    CMDialog1.Filter="All Files|*.*|(*.EXE)|*.EXE"
    CMDialog1.FilterIndex=2
    CMDialog1.DialogTitle="Open file(*.EXE)"
    CMDialog1.Action=1
    If CMDialog1.FileName= " " Then
        MsgBox("No file selected",37,"Checking")
    Else
     'Call a procedure called OpenFile to open the file
        OpenFile(FileName)
    End If
End Sub
```

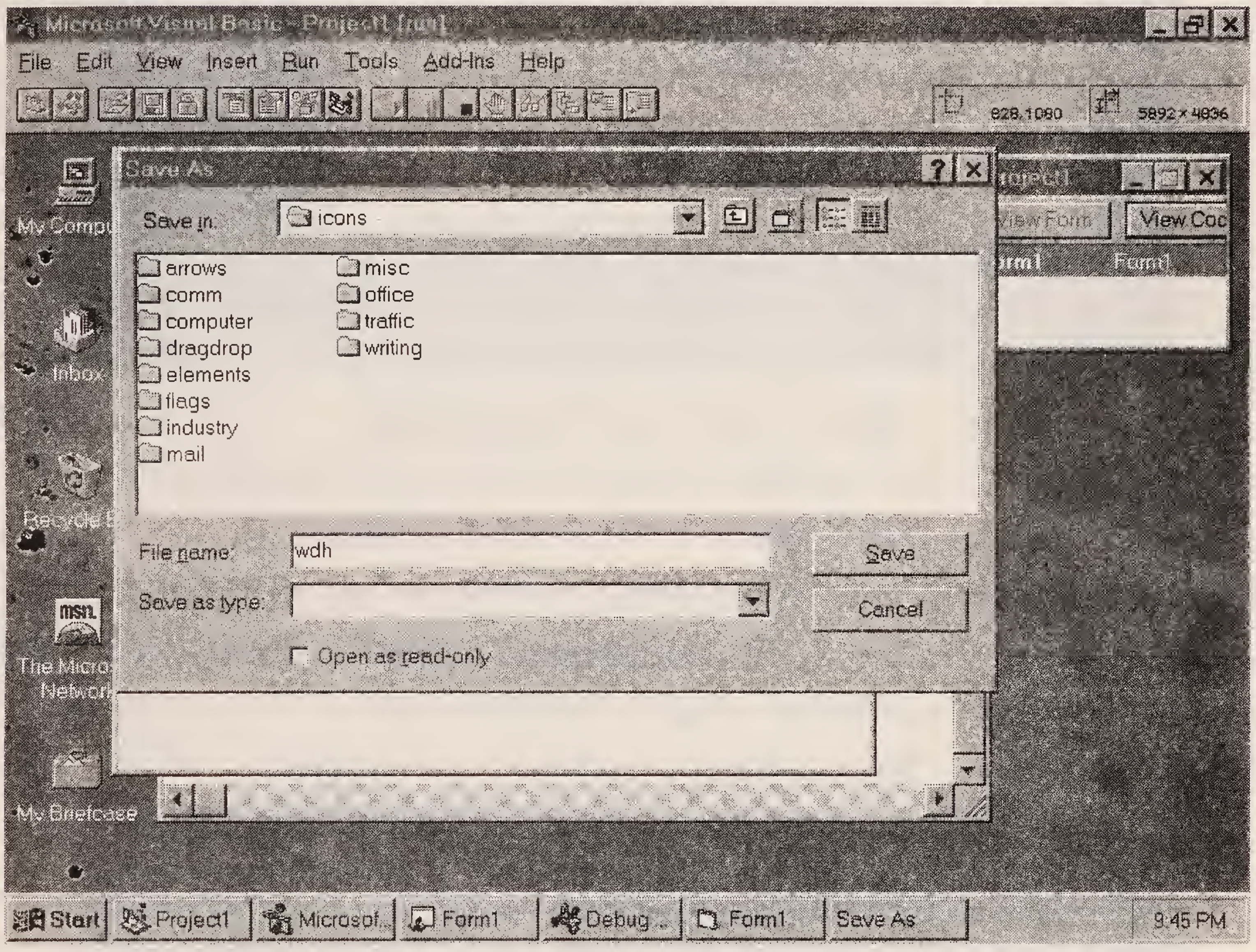
上面程序段前面几句是为 Open 对话框设置外观，后面是用 If 语句判断是否有文件被选择，如果没有文件被选择，则会出现一消息框；如果有文件被选择，则调用打开文件的子过

程 OpenFile()，有关打开文件等操作将在第十章介绍。

8.5.2　Save As 对话框

Save As 对话框的属性基本上和 open 对话框一样，另外还有一个 DefaultExt 属性，它设置为由1至3个字符组成的字符串，用于指定没有文件扩展名时的缺省扩展名。在 Save As 对话框中 DialogTitle 属性的缺省值为 Save As，而不是 Open。

当用户把公用对话框的 Action 属性设置为2时，用户就可以启用如图8.27所示的 Save As 对话框。


Microsoft Visual Basic - Project1 [run]
File Edit View Insert Run Tools Add-Ins Help
828,1080
5892 x 4836
Save As
Project1
View Form
View Code
Form1
My Computer
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
arrows
comm
computer
dragdrop
elements
flags
industry
mail
misc
office
traffic
writing
Save as type:
Save
Cancel
Open as read-only
Start
Project1
Microsof...
Form1
Debug...
Form1
Save As
9:45 PM

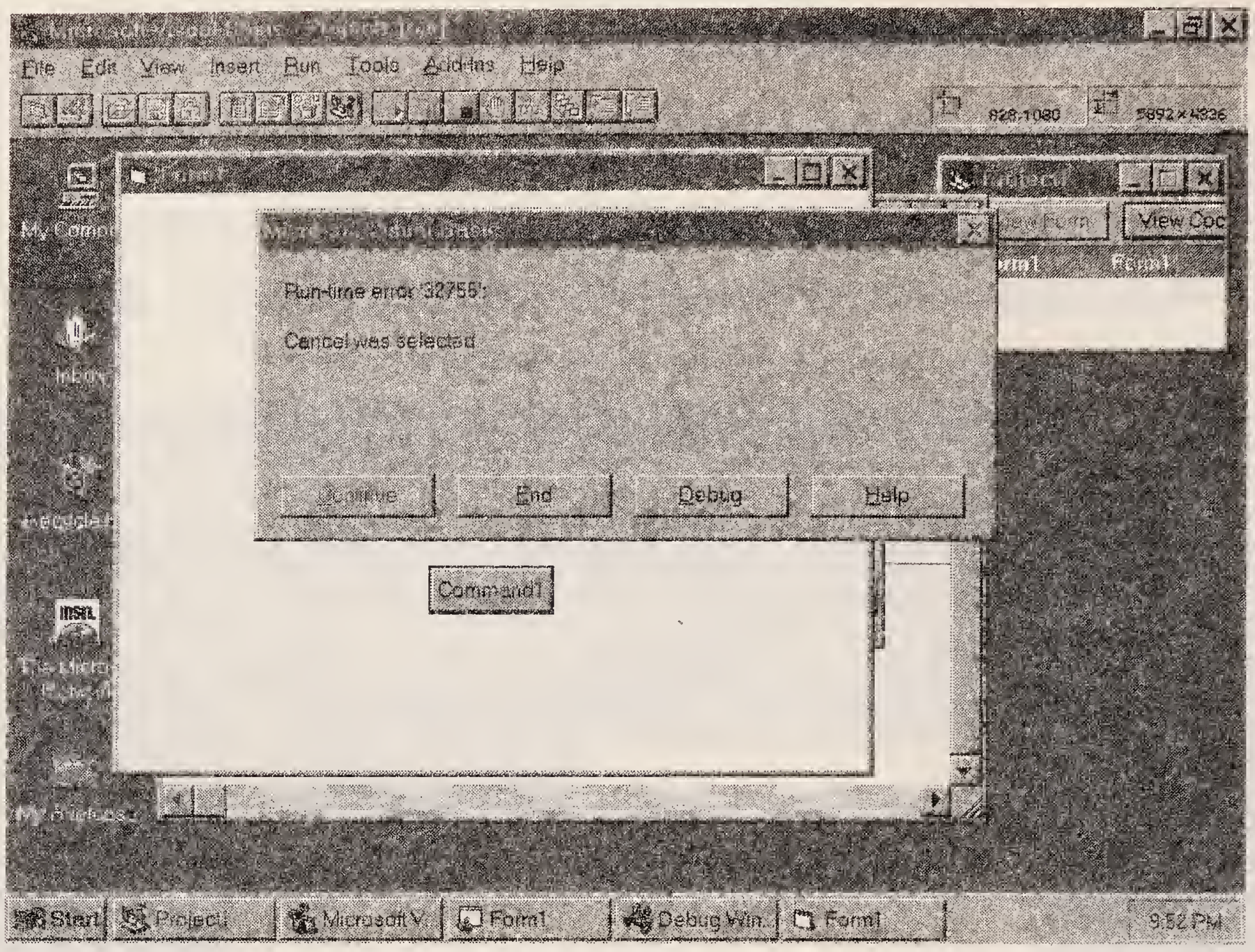
Save in: icons

File name: wdh

图8.27　Save As 对话框

下面一段程序显示一个 Save As 对话框，它假定公用对话框为 CMDialog1。为了说明用户选择了文件，应把 Cancel Error 属性设置为 True，当用户单击 Cancel 按钮时会产生错误信息，见图8.28。

```
CMDialog1.CancelError=True
CMDialog1.DefaultEXT="TXT"
CMDialog1.FilName="WDH.TXT"
CMDialog1.Filter="Text Files(*.TXT)|*.TXT|All Files(*.*)|*.*"
CMDialog1.FilterIndex=1
CMDialog1.Flags=VBOFNOVRWRITEPROMT or VBOFNPATHMUSTEXIST
CMDialog1.Action=2
```


File
Edit
View
Insert
Run
Tools
Add-Ins
Help
828,1080
5892 x 4836
Form1
Project1
View Form
View Code
Form1
Continue
End
Debug
Help
Command1
My Computer
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Start
Project1
Microsoft V...
Form1
Debug Win...
Form1
9:52 PM

Run-time error '32755':

Cancel was selected

图8. 28　单击 Cancel 产生的错误信息

8. 5. 3　Color 对话框

Color 对话框除了有同 Open 对话框相同的 CancelError 和 Flags 等属性以外还有一个 Color 属性，它用来设置初始颜色，使用完对话框后，读此属性可以确定用户选中的颜色。如果设置或读取 Color 属性，必须将 Flags 设置为 VbCCRGBINIT。

当用户将3赋给 Action 属性，启用如图8. 29所示的 Color 对话框。

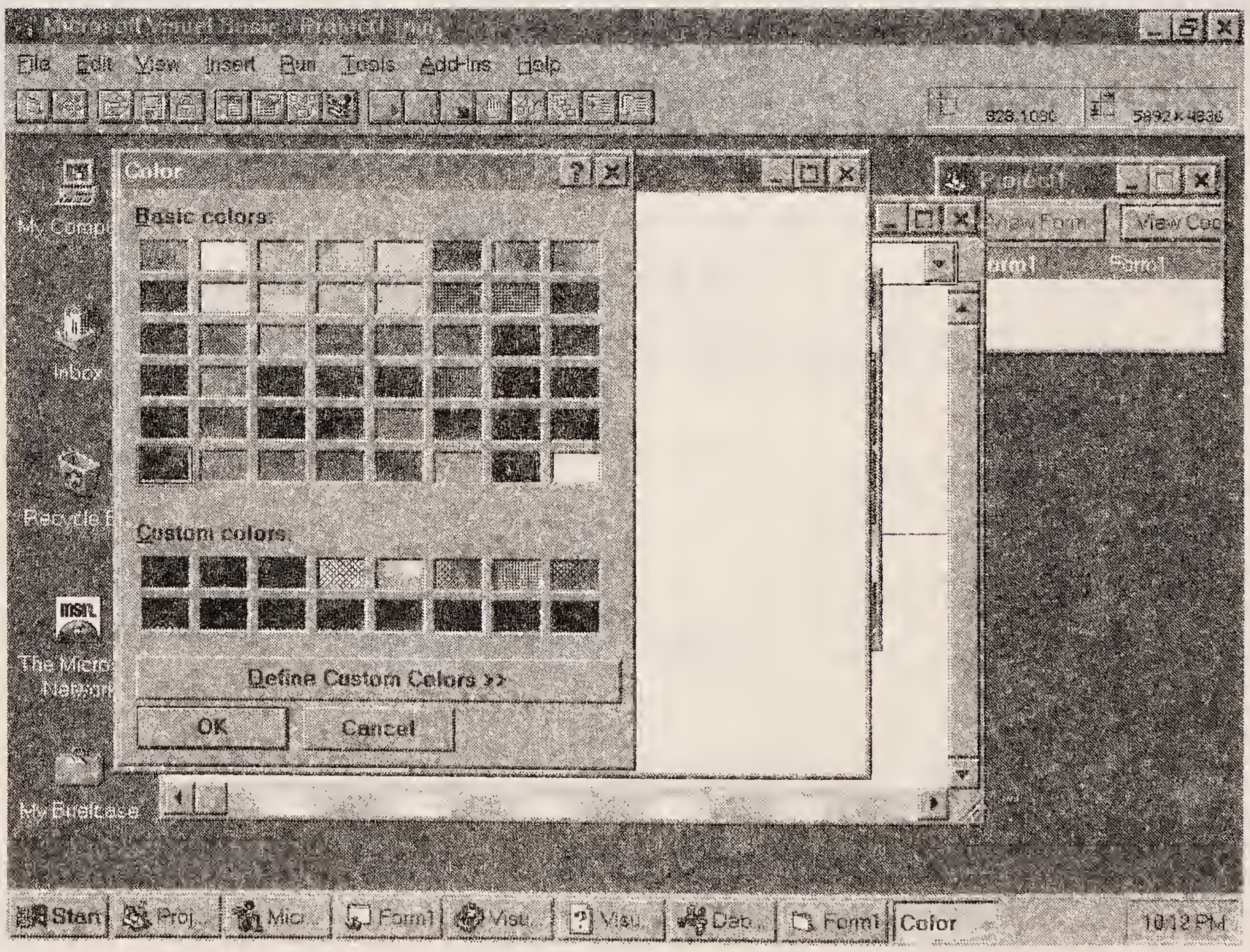

File
Edit
View
Insert
Run
Tools
Add-Ins
Help
828,1080
5892 x 4836
Color
Basic colors:
Custom colors:
Define Custom Colors >>
OK
Cancel
Project1
View Form
View Code
Form1
My Computer
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Start
Proj...
Micr...
Form1
Visu...
Visu...
Deb...
Form1
Color
10:12 PM

图8. 29　Color 对话框

下面一段程序显示一个 Color 对话框，它假定公用对话框的 Name 属性为 CMDialog，并使用选中的颜色设置窗体 Form1的背景颜色。

```
CMDialog. Floags=VbCCRGBINIT
CMDialog. Color=BackColor
CMDialog. Action=3
Form1. BackColor=CMDialog1. Color
```

Color 对话框可以有4个标志来操作。很多情况下，除了 VbCCRGBinit 标志外，并不用其它标志。然而，用户可以用 VbCCFullOpen 自动打开 Color 对话框，并且扩展包括所有已有的颜色定义控制。或者如果想限制选择为标准颜色，则使用 VbCCPreventFullOpen，这将使 Define Custom Colors 按钮无效。第四个标志将使 Help 按钮显示在对话框中。有关对话框的标志值请查看 Visual Basic 4. 0参考手册。

8.5.4 Print 对话框

Print 对话框除了有 Flags、DialogTitle、CancelError、HelpFile 等属性外还有以下属性：

(1) Copies 属性：设置或返回要打印的份数，若设置 VbPDUSEDEVMODECOPIES 标志，则此属性总是返回1。

(2) Max，Min 属性：决定在打印时允许的最大和最小尺寸。

(3) FromPage，ToPage 属性：为 From 和 To 文本框设置或返回值，它决定打印起始页和终止页，只有在设置 VbPDPageNums 时，该属性才有效。

下面的程序段将显示一个打印对话框，见图8. 30。

```
FirstPage=1
LastPage=50
CMDialog1. CancelError=True
CMDialog1. Copies=1
CMDialog1. Min=FirstPage
CMDialog1. Max=LastPage
CMDialog1. Flags=VbPDUSEDEVMODECOPIES or VbPDNOSELECTION
CMDialog1. Action=5
```

注意启动 Print 对话框并没有实际启用打印过程，为启用打印过程必须编写代码用来写至打印机对象。

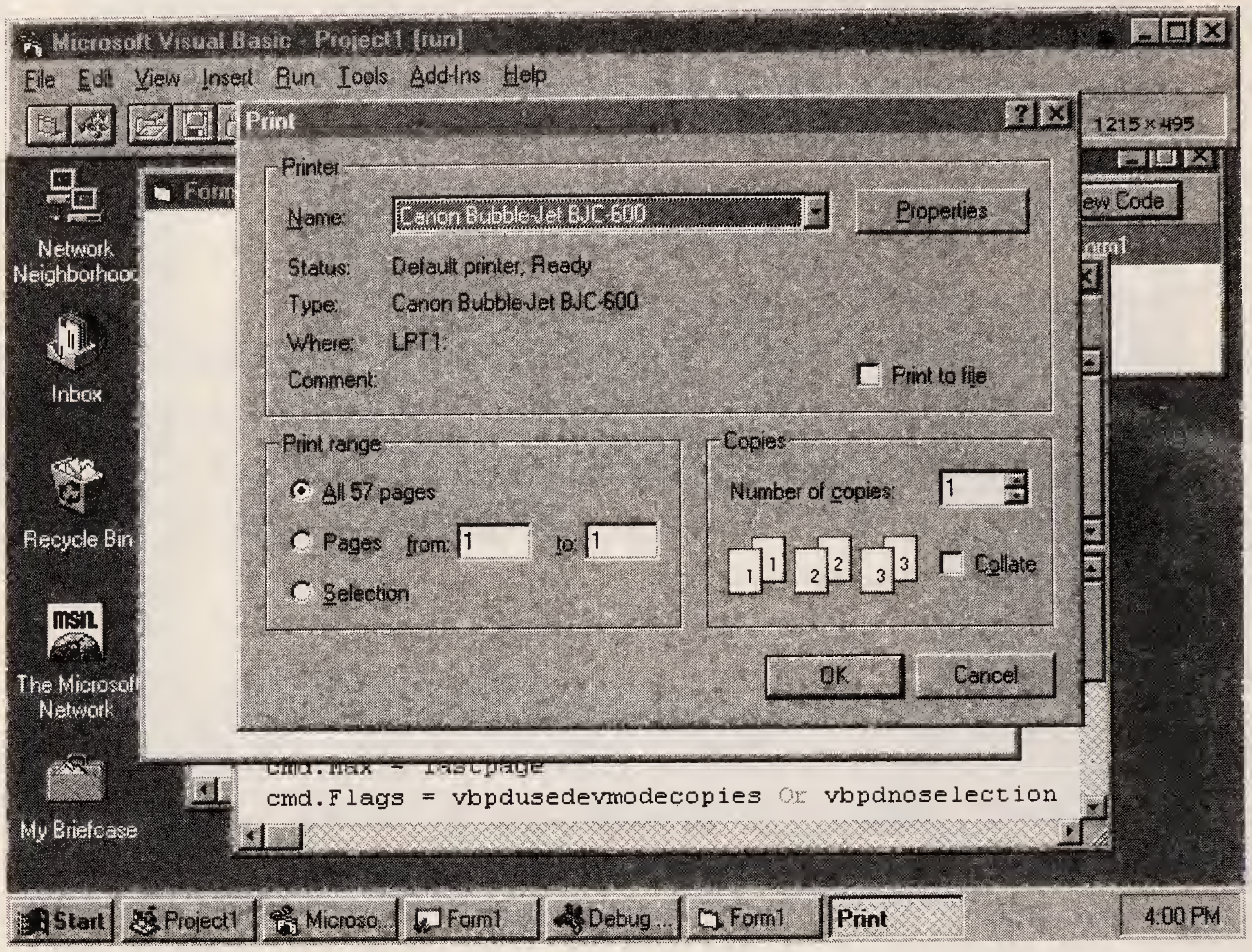
8.6 打印操作

8.6.1 与打印操作有关的方法和属性

1. Print 方法

Print 方法可将文本字符显示在窗体上，将图像显示到图片框(picture box)中。它能够装入或拷贝位图和信息，或打印文本输出到打印机，Print 方法的语法格式为：

```
object. Print[{Spc(n)|Tab(n)}][expressionList]
```


Microsoft Visual Basic - Project1 [run]
File Edit View Insert Run Tools Add-Ins Help
Print
1215 x 495
Network Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
View Code
Properties
OK
Cancel
Start
Project1
Microso...
Form1
Debug...
Form1
Print
4:00 PM

Printer
Name: Canon Bubble-Jet BJC-600
Status: Default printer; Ready
Type: Canon Bubble-Jet BJC-600
Where: LPT1:
Comment:
☐ Print to file

Print range
● All 57 pages
○ Pages from: 1 to: 1
○ Selection

Copies
Number of copies: 1
☐ Collate

```
cmd.Max = lastpage
cmd.Flags = vbpdusedevmodecopies Or vbpdnoselection
```

图8.30 Print 对话框

其中 object 代表要打印到的对象。例如窗体、图片框或打印机等，缺省对象为当前窗体。Spc(n)是可选的函数名，可以在打印输出部分中插入 n 个空格，可多次使用。Tab(n)也是可选的的函数名，用来在打印表达式之前的第 n 列进行标志，可多次使用。expressionList 代表要打印的串表达式，若省略，Print 将输出空行，多项表达式以分号结尾，则意味着光标紧随着上一个字符之后。串表达式以逗号结尾，则意味着光标在同一行的下一个打印区开始。如果串表达式结尾没有分号和逗号，则下一个字符将从下一行开始。

2. CurrentX，CurrentY 属性

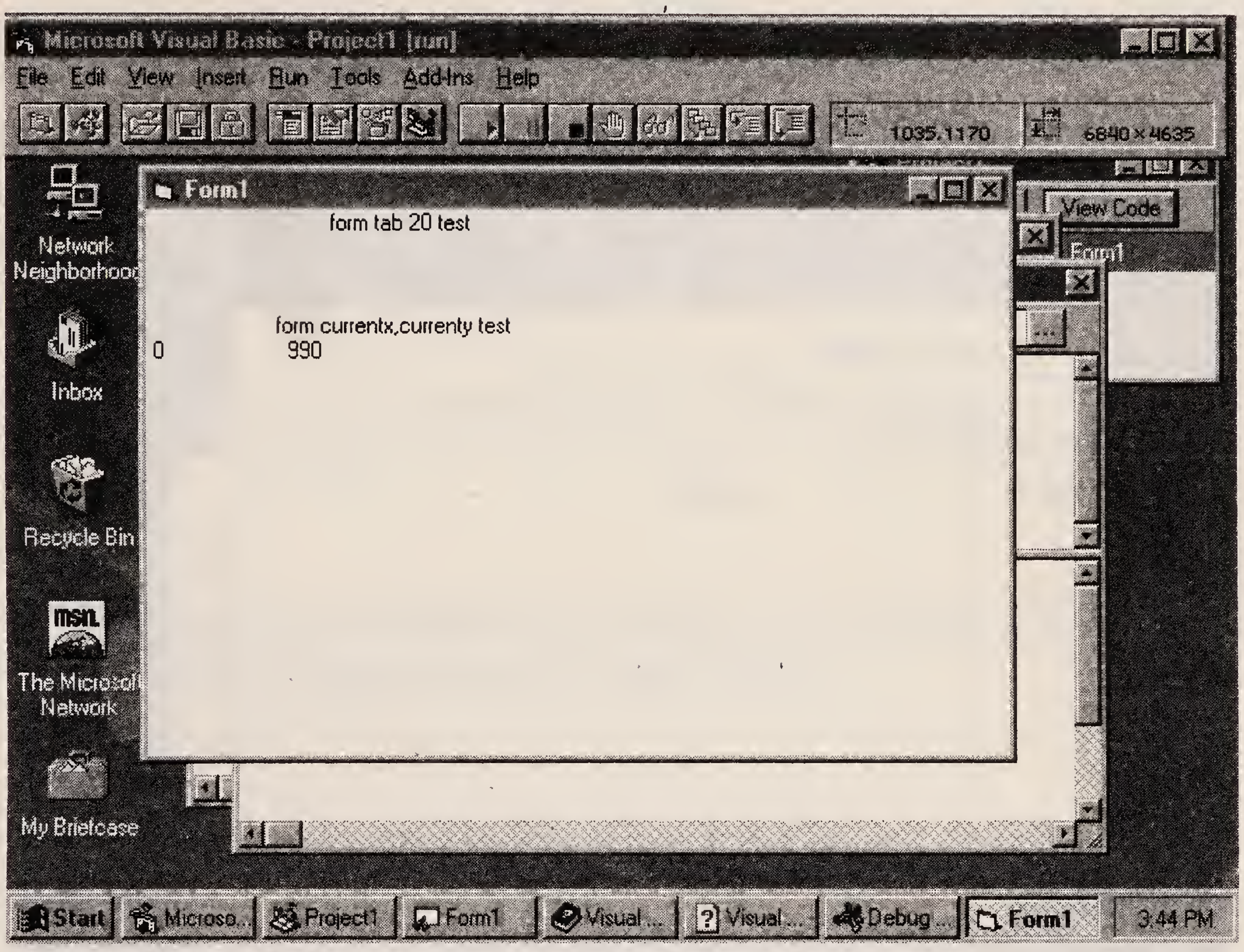
这两个属性确定下一次打印或绘画的水平((CurrentX)和垂直(CurrentY)坐标，语法格式如下。

```
Object. Printer. CurrentX=[X]
Object. Printer. CurrentY=[Y]
```

坐标是从对象的左上角开始计算的，对象的右边界处的 CurrentX 为0，对象的顶边界处的 CurrentY 为0，坐标以像素为单位表示。例如下面一段代码就是使用 CurrentX 和 CurrentY 来确定打印位置。

```
Sub Form _ Click()
    Forml. Print Tab(20);"from tab 20 text"
            Form1. CurrentX=1000
            Form1. CurrnetY=800
    Form1. Print "from CurrentX,CurrentY text"
    Form1. Print CurrentX,CurrentY
End Sub
```

本例的第一条语句是在窗体的第20个位置处(Tab(20))输出“from tab 20 test”字符串。第2,3语句重新设置窗体的 CurrentX 和 CurrentY 属性,所以第4行用新的坐标输出“form-CurrentX,CurrentY test”。最后一行语句本想输出新的坐标值,但是坐标好像被重新设置。注意,如果设置完坐标以后再用 Print 方法,CurrentX 和 CurrentY 会被设置到下一个输出位置(如果再遇到 Cls 方法,则会重新设置为(0,0)位置)。下图是程序的运行结果:



```
                    form tab 20 test


             form currentx,currenty test
0              990
```

Microsoft Visual Basic - Project1 [run]
File Edit View Insert Run Tools Add-Ins Help
1035,1170
6840 x 4635
Form1
View Code
Form1
Network Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Start
Microso...
Project1
Form1
Visual ...
Visual ...
Debug ...
Form1
3:44 PM

图8.31 程序运行结果显示

3. TextHeight 与 TextWidth 方法

TextHeight 方法是返回一文本串按当前字体打印的高度。TextWidth 方法是返回文本字符串的宽度,用对象的当前字体打印,它们的语法格式为:

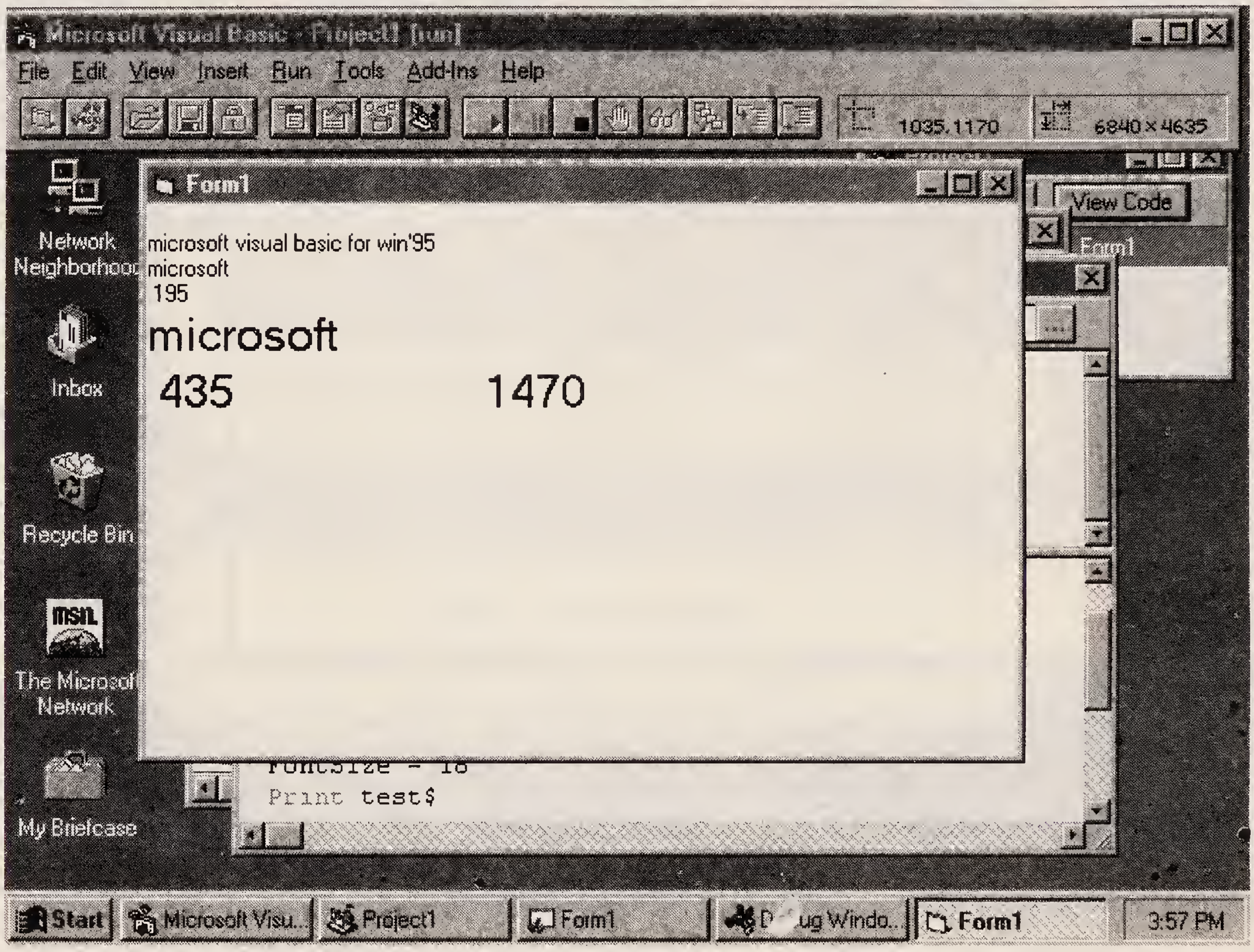
```
Object. TextHeight(String expression)
Object. TextWidth(String expression)
```

Object 代表用来计算文本高度的字体尺寸的对象(窗体、图片框或 Printer 对象)。String expression 代表要确定其高度的文本字符串。下面是一段利用这两种方法的一段代码:

```
Sub Form _ Click()
  CurrentY=TextHeight("Microsoft") * 6
  CurrentX=0
Print "Microsoft Visual Basic for Winows95"
  Test $ ="Microsoft"
  print test $
  print TextHeight(Test $ ),TextWidth(Test $ )
  FontName="System"
  FontSize=18
```

```
    print Test $
    print TextHeight(Test $ ),Textwidth(Test $ )
  End Sub
```

第一个语句是用“Microsoft”字符高度乘以6作为 Print 位置的纵坐标，横坐标为0。第4行的 Print 语句的打印会从第7行第1列开始。第6条语句显示了“Microsoft”字符的高度和宽度，第7、8条语句设置字符的字形和大小，结果第9条语句显示不同字形和大小的“Microsoft”字符串，同时第10条语句输出字符的高度值和宽度值。图8.32是程序的运行结果。



```
microsoft visual basic for win'95
microsoft
 195
microsoft
 435                1470
```

```
FontSize = 18
Print test$
```

Microsoft Visual Basic - Project1 [run]
File Edit View Insert Run Tools Add-Ins Help
1035,1170
6840 x 4635
Form1
View Code
Network Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Start
Microsoft Visu...
Project1
Form1
Debug Windo...
Form1
3:57 PM

图8.32 程序运行结果显示

8.6.2 直接将信息送往打印机

Visual Basic 允许用户通过用 Print 方法将有关信息直接输送到打印机上。为了把信息直接输送到打印机上，用户只需把 Print 方法的对象选为 Printer(打印机)即可。例如，下面一条语句将把字符串“Microsoft Visual Basic 4.0”输送到缺省的打印机上：

```
Printer. Print "Microsoft Visual Basic 4.0"
```

在打印之前，用户能够用 Printer 对象修改字体属性，例如下面代码将用“Roman”字体和14点的大小打印“Microsoft Visual Basic 4.0”字符串。

```
Printer. FontBame="Roman"
Printer. FontSize=14
Printer. Print "Microsoft Visual Basic 4.0"
```

Printer 对象有许多属性和方法使用户能够控制打印的各个方面情况。不像窗体和控件，用户不能通过属性窗口来设置 Printer 的属性，每一个属性都必须用程序代码在运行时设置。

仅仅将信息送到带有Print方法的Printer对象，对打印文档是不够的。用户将不得不考虑在Printer对象中定位文档的有关处理。如页结束的处理，文档的结束处理等。如果不作其它假定，可以直接使用Printer对象的Print方法在当前位置上打印信息。例如，在打印一行文本之后再打印另一行文本，那么，第二行文本将接在第一行文本的后面。如果想要在指定位置打印信息，可以对二个Printer对象的属性进行设置以控制打印位置：CurrentX和CurrentY。

当运用Printer. NewPage语句开始一新页时，CurrentX和CurrentY属性都被设置为0，使打印定位于打印机的左上角。一旦用户开始新的一页，就可以用NewPage方法。在结束文档打印时也应使用NewPage方法。除此之外，在想结束文档打印时，还必须使用EndDoc方法。因此，下面二句在打印完成时必须执行：

```
Printer. NewPage
Printer. EndDoc
```

如果用户想打印“Text 1”文本框内所有文档，可以使用下列语句：

```
Printer. Print "Text1. Text"
Printer. NewPage
Printer. EndDoc
```

第一条语句将整个文档内容送到Printer对象中。第二语句结束当前页进入下一页。第三条语句中的EndDoc方法结束文档并将它送到打印机中。

8. 6. 3 窗体输出

另一种打印机输出的方法就是将欲输出的信息先送至窗体，然后用PrintForm方法将整个窗体从打印机输出。PrintForm方法的语法格式为：

```
formname. PrintForm
```

FormName代表将要打印的窗体，若被省略，则打印当前的窗体。PrintForm在运行时也打印加在窗体或图形控件的图像（如果当绘出图形时AutoRedraw属性为True）。PrintForm所使用的打印机由Windows 95 Control Panel（控制面板）决定。下面代码是使用PrintForm方法的一例：

```
Sub Form _ Click()
    fontname = "Roman"
    fontsize = 14
    Form1. Print "Microsoft Visual Basic for Windows 95"
    PrintForm
End Sub
```

第九章　图形及绘图操作

用户在设计程序时，所处理的内容不一定都是文字信息。有时要绘制一些图形，有时又要为文件增加一些图像，这样就得运用绘图以及图形操作。一个图文并茂的文件比一个只有文字信息的文件能表达更多的信息，且更易阅读，更直观。

9.1　图像工具

Visual Basic 为用户提供了许多图像控件，便于用户进行图形及绘图操作时使用。其中最主要的是 Picture Box（图片框）、Image（图像工具）、Line（画线工具）以及 Shape（形状）控件。下面我们就介绍一下这些工具。

9.1.1　Picture Box（图片框）工具

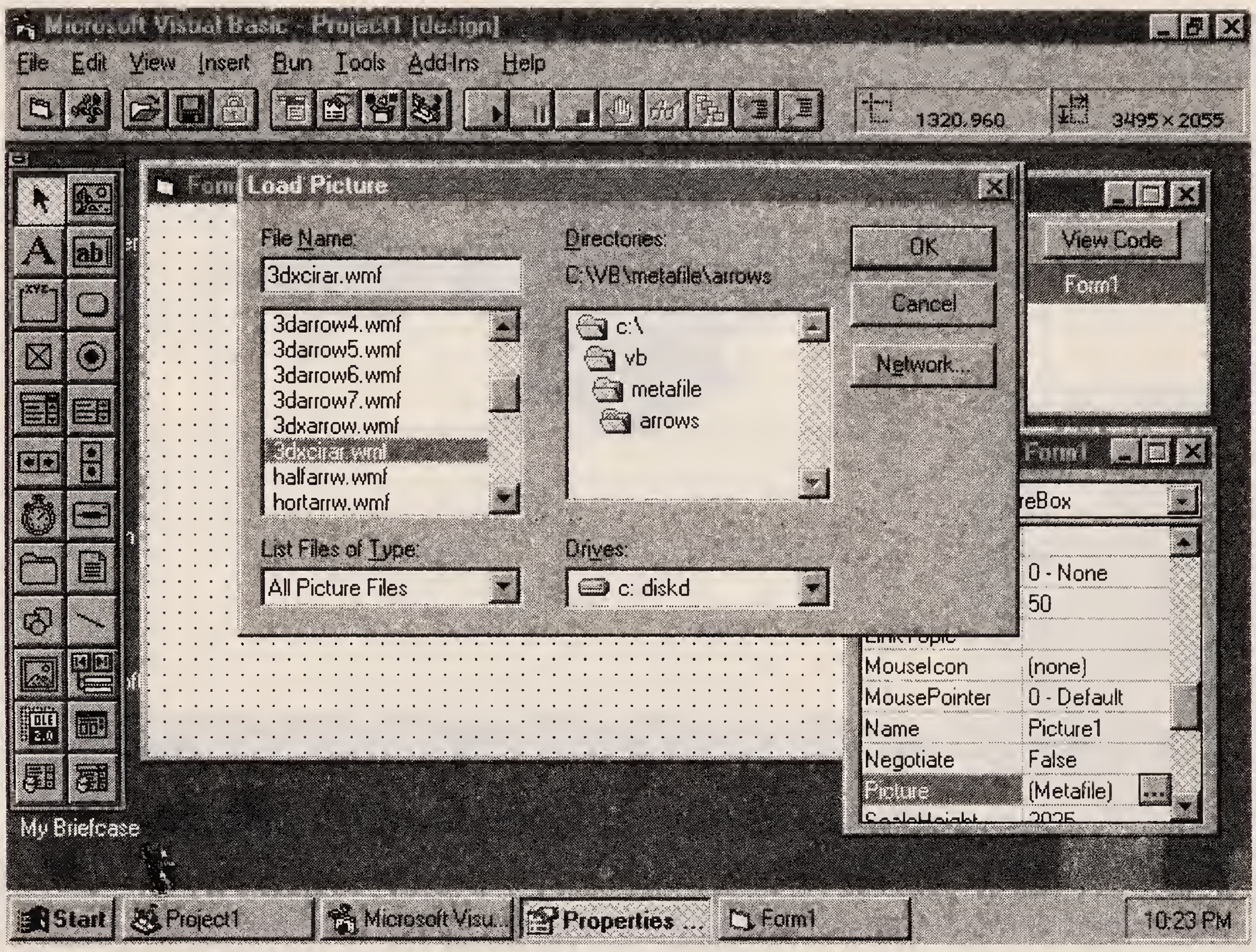
图片框可以显示 bitmap（位图）、icon（图标）以及 metafile（元文件）中的图形。过程是首先选择工具框中的图片框工具，在 Form 窗口内画出一个矩形框体。然后选择 Properties 窗口中的 Picture 属性，这时，在 Picture 属性的设置框里会出现一带有省略号图标的按钮。单击这一按钮，即会出现 Load Picture 对话框，如图9.1所示。用户可以从对话框中选择 bitmap、icon 或 metafile 文件夹，然后在路径对话框中双击这一文件即在文件夹下面列出各文件名。单击某一文件名，在左边的 FileName 列表框内即会出现这一文件中所包含的各种图像名称，在 FileName 列表框内选择想要的图形，然后单击 OK 按钮。如果想要取消刚才选择的这些图形，只需要单击 Cancel 按钮就可以了。单击 OK 按钮后，对话框即消失了，并且马上在我们前面画的框体内出现一个图形，如图9.2所示。

有时，我们想用一个图片框把一个图标完整地框起来。一般来说，一个图标非常小。在画一个图片框时，我们一般很难把框体画的边界与图标的边界完全重合。这时，我们就需要对图片框的Autosize属性进行设置。即在图片框为当前活动控件的条件下（如图片框不为当前活动控件，在 Form 窗口中单击这一控件，即可使其成为当前活动控件），选择属性窗口中的 Autosize属性，将其设置为 True。这时这一图片框就可以自动改变大小来适应实际图形的大小。如果将 Autosize 属性设置为 False，则图片框体大小就不能自动改变来适应图形的形状了。

如果图片框的 Autosize 属性为 False，图片框就不能自动改变大小来适应其中的图形。而当图形框中的图形为 BMP（位图）或 ICO（图标）文件中的图形时，它们也不会自动重设大小来填充图片框。如果图片框中的图形为 metafile（元文件）中的图形时，则这些图形会自动重设大小来填充图片框。元文件的组成是由线、圆及其它图形元素构成的图形。对元文件重设图形大小时，这一些图形元件将重画，以维持原来的图形品质。

例如，我们在 Form 窗口内画一图片框，其形状是一任意大小的矩形，然后将其 Autosize 属性设置为 False，这时此图片框的大小将不会自动改变。当然我们还可以通过其边框上的

八个小黑框来改变其大小。然后选择 Properties 窗口中的 Picture 属性，在打开的对话框中选择图9.2中的图形，然后单击 OK 按钮。这时在图片框中的图形虽与图9.2中属于同一个图形，但形状大小已经改变了，这是因为图形自动改变大小来适应图片框，如图9.3所示。


Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
1320, 960
3495 × 2055
Load Picture
OK
Cancel
Network...
View Code
Form1
My Briefcase
Start
Project1
Microsoft Visu...
Properties ...
Form1
10:23 PM

File Name: 3dxcirar.wmf

3darrow4.wmf
3darrow5.wmf
3darrow6.wmf
3darrow7.wmf
3dxarrow.wmf
3dxcirar.wmf
halfarrw.wmf
hortarrw.wmf

Directories: C:\VB\metafile\arrows

c:\
vb
metafile
arrows

List Files of Type: All Picture Files

Drives: c: diskd

	0 - None
	50
MouseIcon	(none)
MousePointer	0 - Default
Name	Picture1
Negotiate	False
Picture	(Metafile)

图9.1　Load Picture 对话框

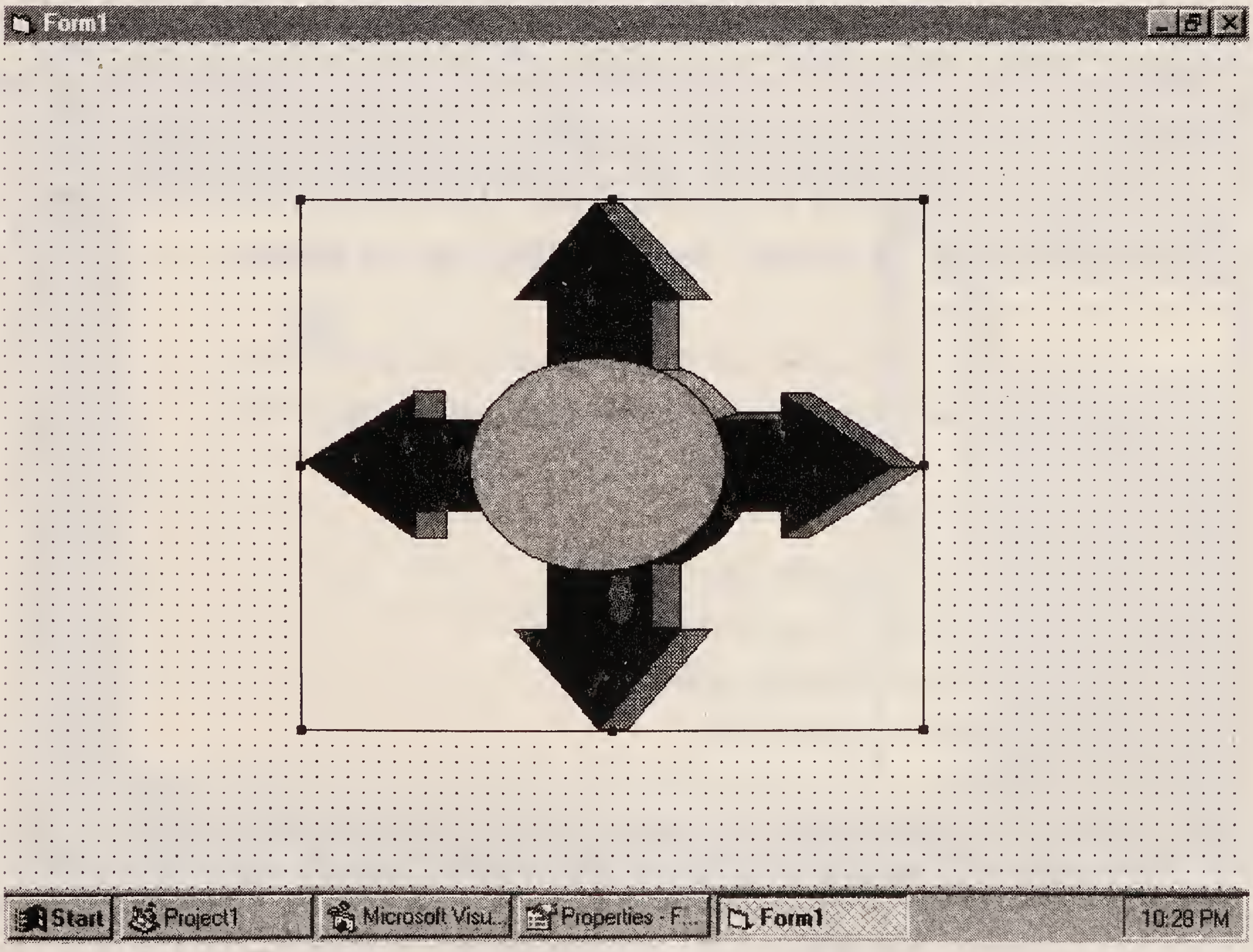

Form1
Start
Project1
Microsoft Visu...
Properties - F...
Form1
10:28 PM

图9.2　图片框中的3dxcirar.wnf 图形

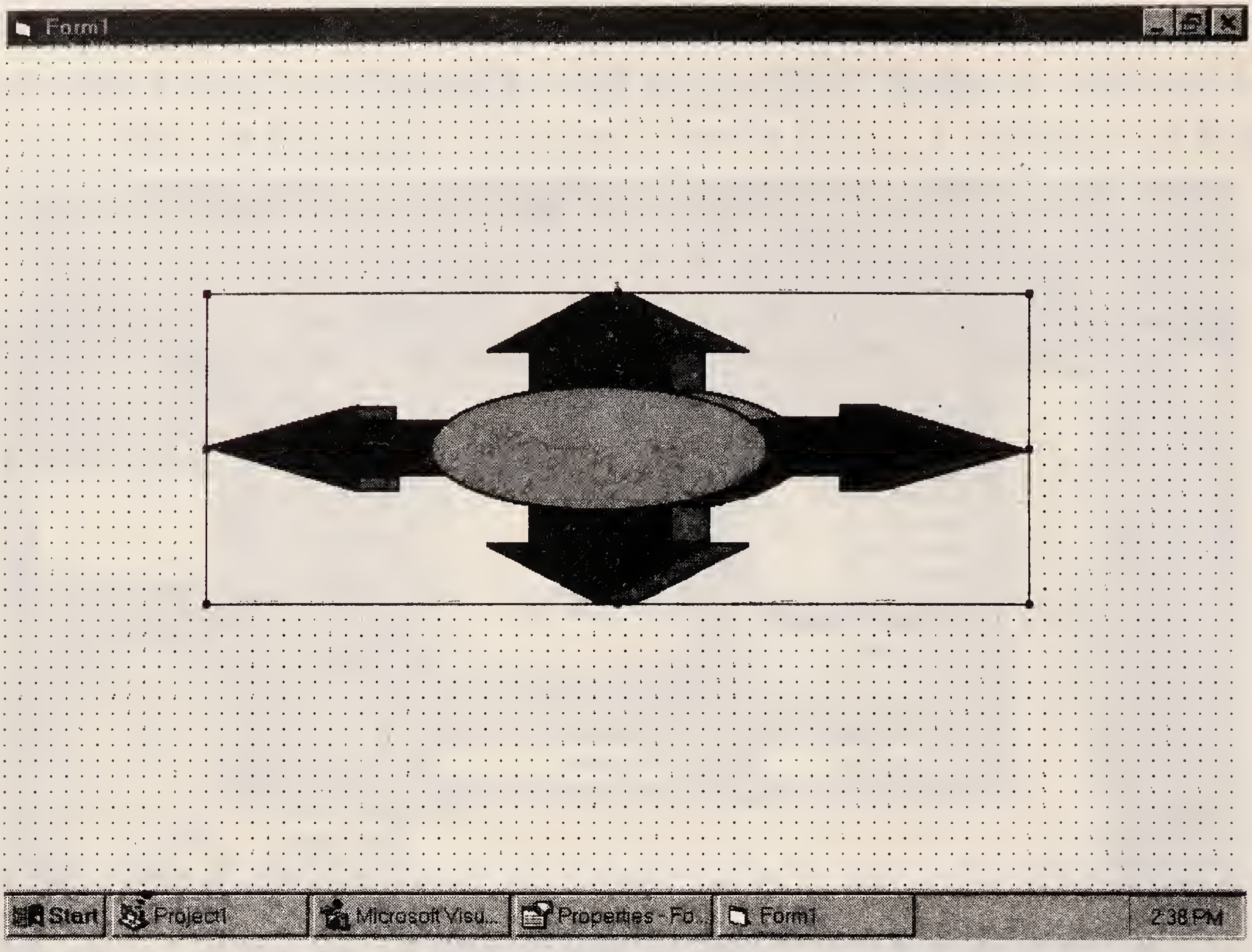


图9.3 改变图片框大小后的3dxcirar 图形

除了直接选择 Properties 窗口中的 Picture 属性给图片框中添加图形外，还可以通过编写代码，在程序运行时在 Form 窗口中的图片框中加入图形。

下面，我们将举一个例子。在 Form 窗口中画出三个图片框，然后在其中的两个图片框中装上两个图标。当单击窗体时，两个装有图标的图片框将交换图标。其具体过程如下：

首先，建立一个新的工程文件，将 Form 窗口的 Caption 属性设置为 Icons。然后选择工具框中的图片框控件，在 Form 窗口内画出三个图片框，并且将它们的 Autosize 属性全部设置为 True。将它们的 Name 属性选定为缺省名，分别为 Picture1、Picture2和 Picture3。然后双击 Form 窗口或单击 Project 窗口中的 View Code 按钮，输入下列代码。

```
Private Sub Form _ Load()
    Picture1. Picture=Loadpicture("C：\Vb\icons\traffic\trffco1. ico")
    Picture2. Picture=Loadpicture("C"\Vb\icons\traffic\trffco8. ico")
End Sub
Private Sub Form _ Click()
    Picture3. Picture=Picture1. Picture
    Picture1. Picture=Picture2. Picture
    Picture2. Picture=Picture3. Picture
    Picture3. Picture=Loadpicture()
End Sub
```

这段代码的意思是，首先在第一个图片框内装入图标 trffco1，然后在第二个图片框内装入图标 trffco8。如图9.4所示。在单击窗体窗口以后，图片框1和图片框2互换图标，其过程是首先图片框3装入图片框1中的图标 trffco1，然后图片框1中的图标 trffco1被图片框2中的图标 trffco8代替。再把图片框3也就是原图片框1中的图标 trffco1装入图片框2。这样，图片框1

与图片框2中的图标就互换了。如图9. 5所示。图片框3的作用就是在图片框1与图片框2互换图标时起个媒介的作用。当再次单击 Form 窗口时，图片框1与图片框2将再次互换图标，单击一次，互换一次。

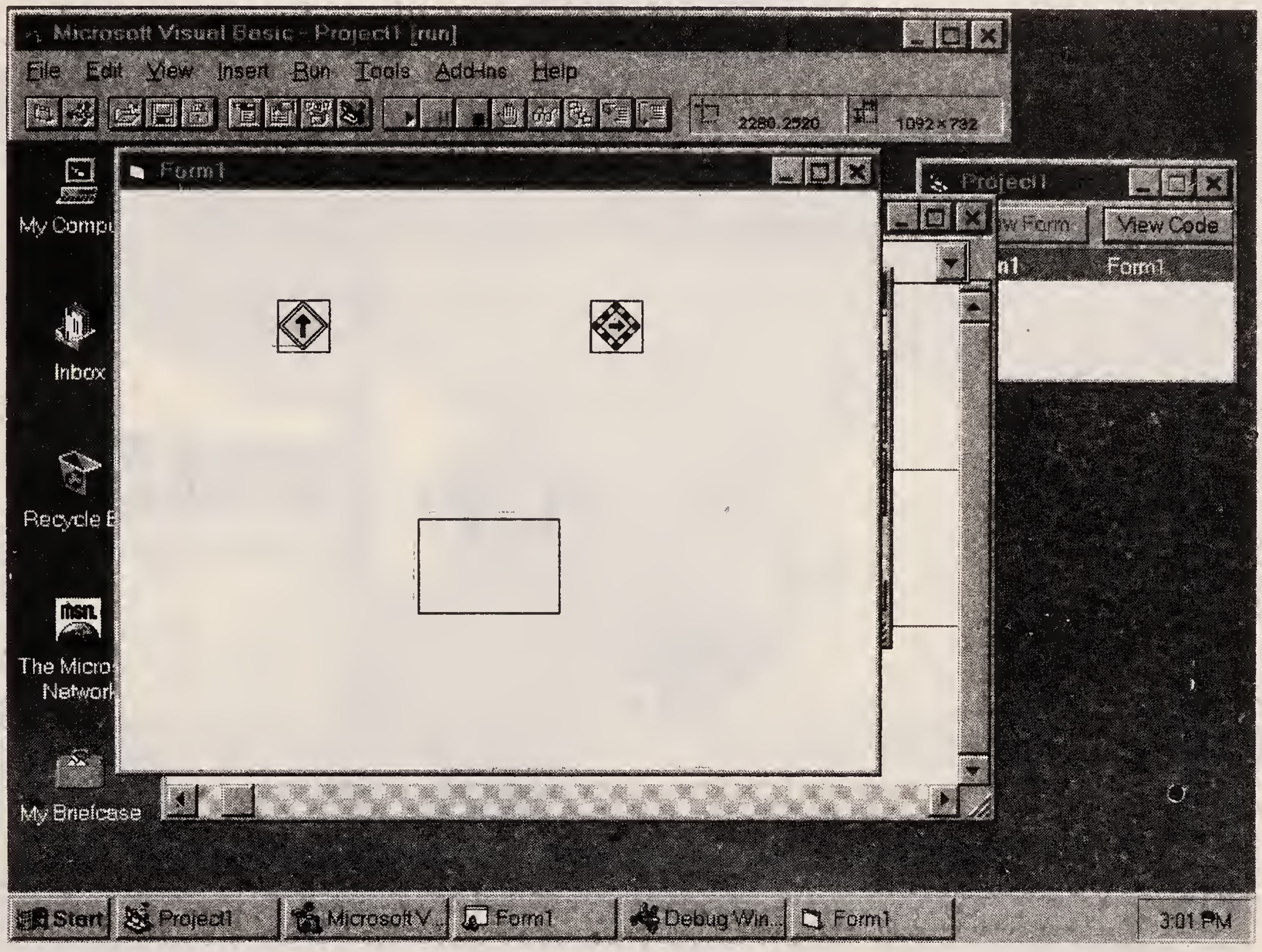

Microsoft Visual Basic - Project1 [run]
File Edit View Insert Run Tools Add-Ins Help
2280,2520
1092×732
Form1
Project1
w Form
View Code
n1
Form1
My Compu
Inbox
Recycle B
The Micro
Network
My Briefcase
Start
Project1
Microsoft V...
Form1
Debug Win...
Form1
3:01 PM

图9. 4　在图片框1与图片框2中装入图标

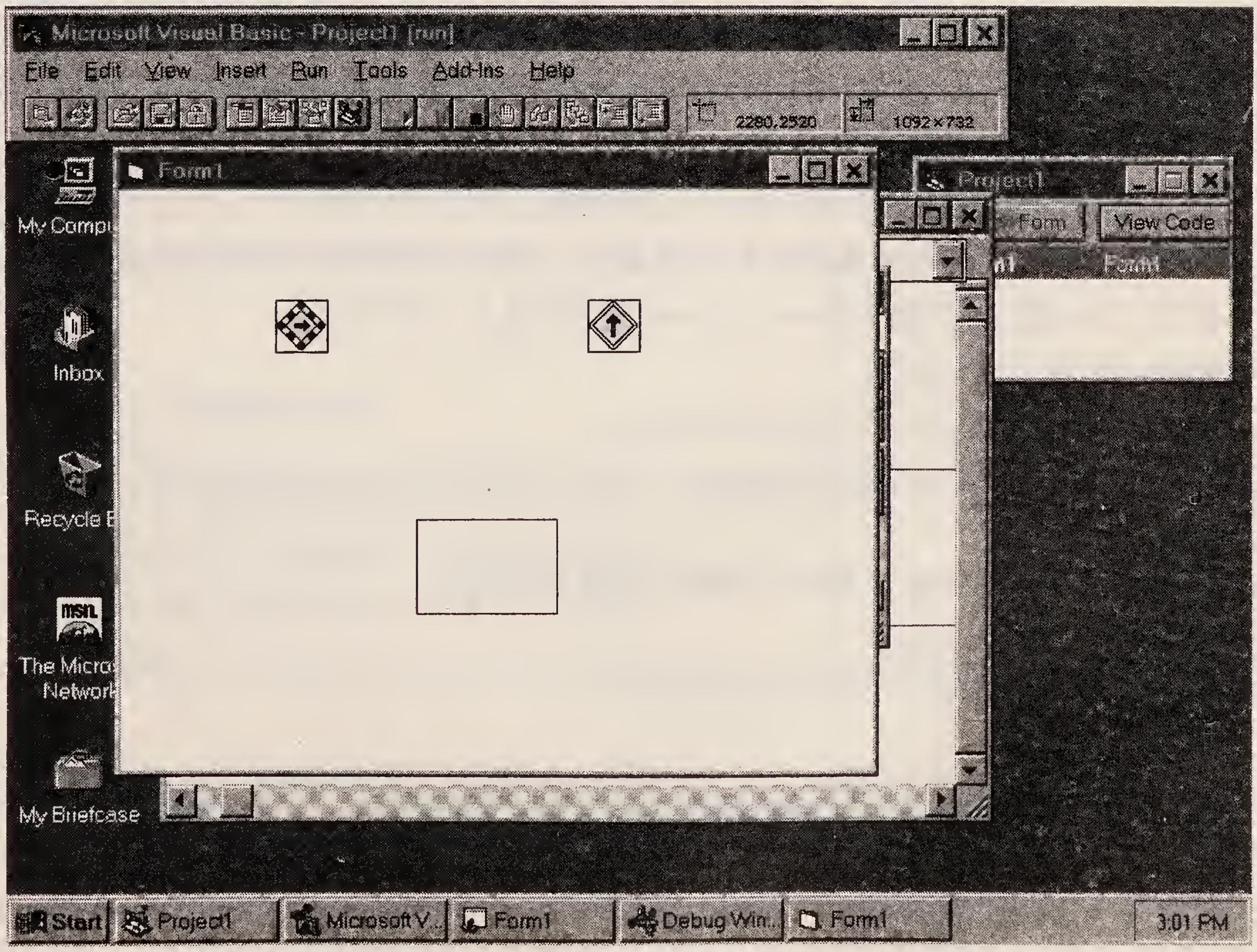
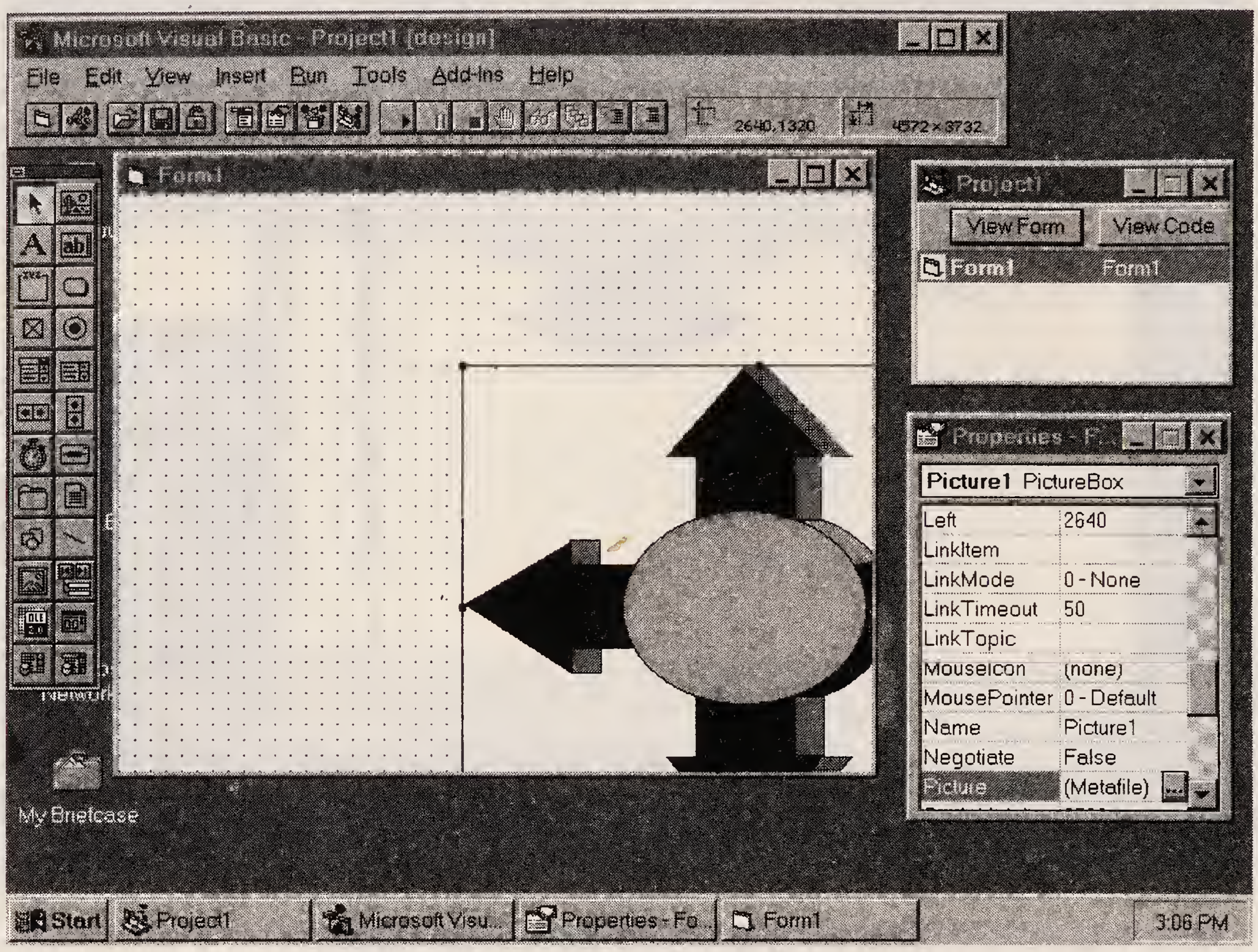

Microsoft Visual Basic - Project1 [run]
File Edit View Insert Run Tools Add-Ins Help
2280,2520
1092×732
Form1
Project1
w Form
View Code
n1
Form1
My Compu
Inbox
Recycle B
The Micro
Network
My Briefcase
Start
Project1
Microsoft V...
Form1
Debug Win...
Form1
3:01 PM

图9. 5　图片框1与图片框2互换图标

我们在图9.2中看到的图形是在 Form 窗口最大化时的图片框中的图形，也就是说这时 Form 窗口覆盖了整个屏幕。如果当我们单击最大化的 Form 窗口的右上角的三个图标中间的那个图标，也就是恢复图标命令时，Form 窗口将恢复到放大前的大小。由于图片框不能重设大小以适应窗口，所以图片框中的图形将被 Form 窗口边界截断，如图9.6所示。


Microsoft Visual Basic - Project1 [design]
File
Edit
View
Insert
Run
Tools
Add-Ins
Help
2640,1320
4572 x 3732
Form1
Project1
View Form
View Code
Form1
Form1
Properties - F...
My Briefcase
Start
Project1
Microsoft Visu...
Properties - Fo...
Form1
3:06 PM

Picture1 PictureBox

Left	2640
LinkItem	
LinkMode	0 - None
LinkTimeout	50
LinkTopic	
MouseIcon	(none)
MousePointer	0 - Default
Name	Picture1
Negotiate	False
Picture	(Metafile)

图9.6　图片框被 Form 窗口边界所截

当然，我们可以通过代码来改变图片框的属性，使之能随着窗体窗口大小的改变而自动改变大小，以适应窗口，其步骤是：

(1) 选择 Picture1控件，在 Form 窗口上随意画一图片框控件。

(2) 选择 Picture1的 Align 属性，将其设置为1，使工具顶端与窗体顶端对齐。

(3) 打开 Code 窗口并选择 Form _ Load 过程，输入下列代码：

```
Private Sub Form _ Load()
        Picture1.Height=Form1.ScaleHeight
        Pictrue1.Picture=LoadPicture("C:\Vb\metafile\arrows\3dxcirar.wmf")
End Sub
```

(4) 在 Proc 列表框中选择 Resize 事件，并输入代码。

```
Private Sub Form _ Resize()
        Picture1.Height=Form1.ScaleHeight
End Sub
```

(5) 运行程序。

(6) 改变窗体大小，我们会发现图片框也随着作了相应的改变。如果装入的是元文件图形，它也将重设大小以匹配新窗口。如图9.7所示。

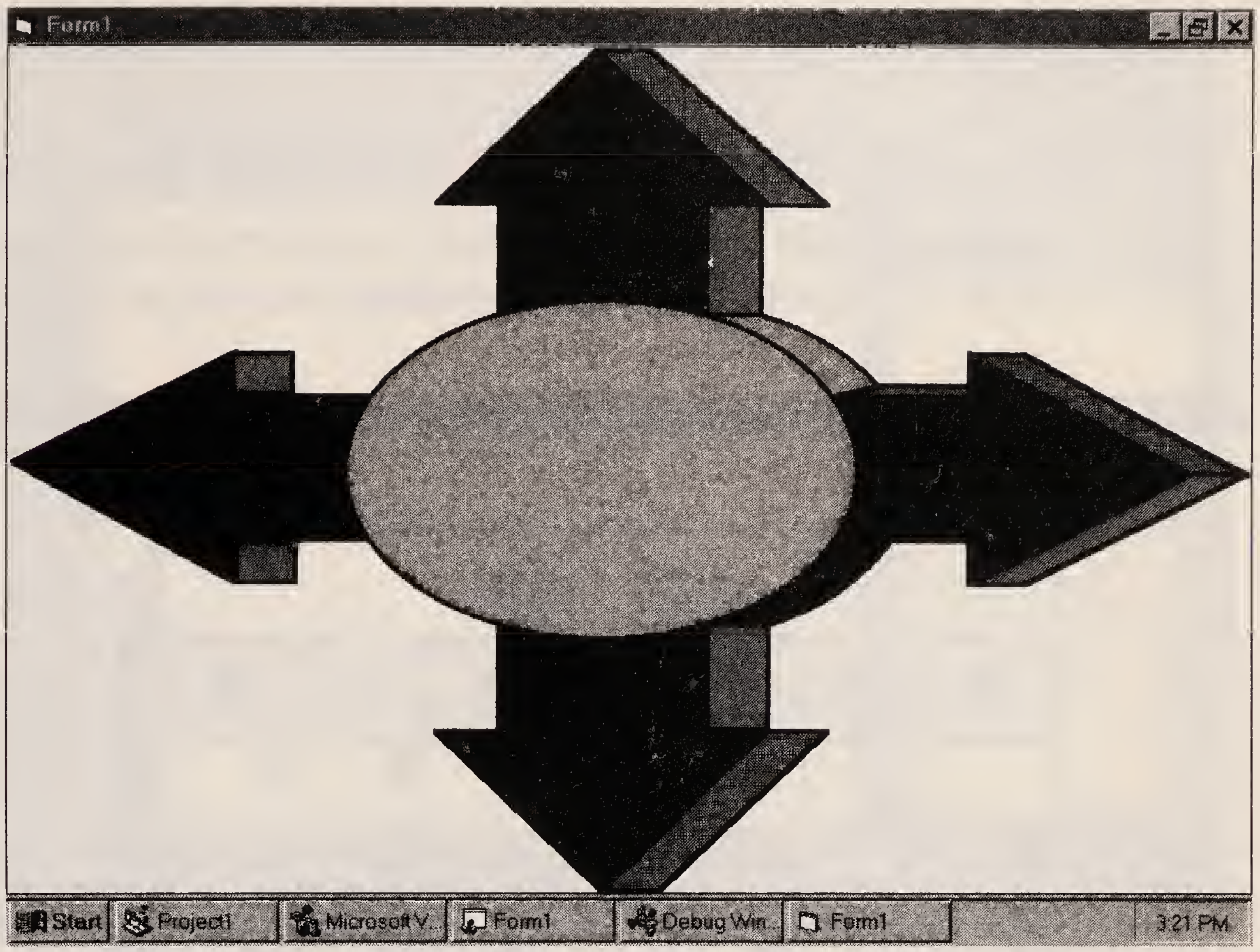


图9.7　图片框重设大小以适应窗体大小

当然，当图片框与窗体同等大小时，我们还可以将图片框的 BorderStyle 属性设置为0，将图片框的边缘线除去。

前面的第二步，选择 Align 属性为1，使图片框与窗体顶端对齐，是为了使图片框的宽度可以自动改变大小。第三、四步是使图片框的高度可随 Form 窗口的高度改变而改变。这样图片框就可以自动改变大小来适应窗体窗口的尺寸了。

图片框还可以显示绘图方法的输出，Print 方法写的文本，以及组合选项按钮等。如果想制作动画和仿真，还可在程序中使用绘图特性和方法。绘图特性和事件对于进行打印操作也是有用的，例如修改一个屏幕窗体的打印格式。一个图片框还可扮作一个 DDE 对话中的客户。

9.1.2　Image（图像框）工具

图像框工具的作用也是在其框体内添加一幅图形。图形可为 bitmap（位图）、icon（图标）或 metafile（元文件）中的图形。添加图形过程为：

（1）选择 File 菜单中的 New Project 命令。

（2）将新的 Form 窗口的 Caption 属性设置为 Image，其控制名（Name）属性的缺省值为 Form1。

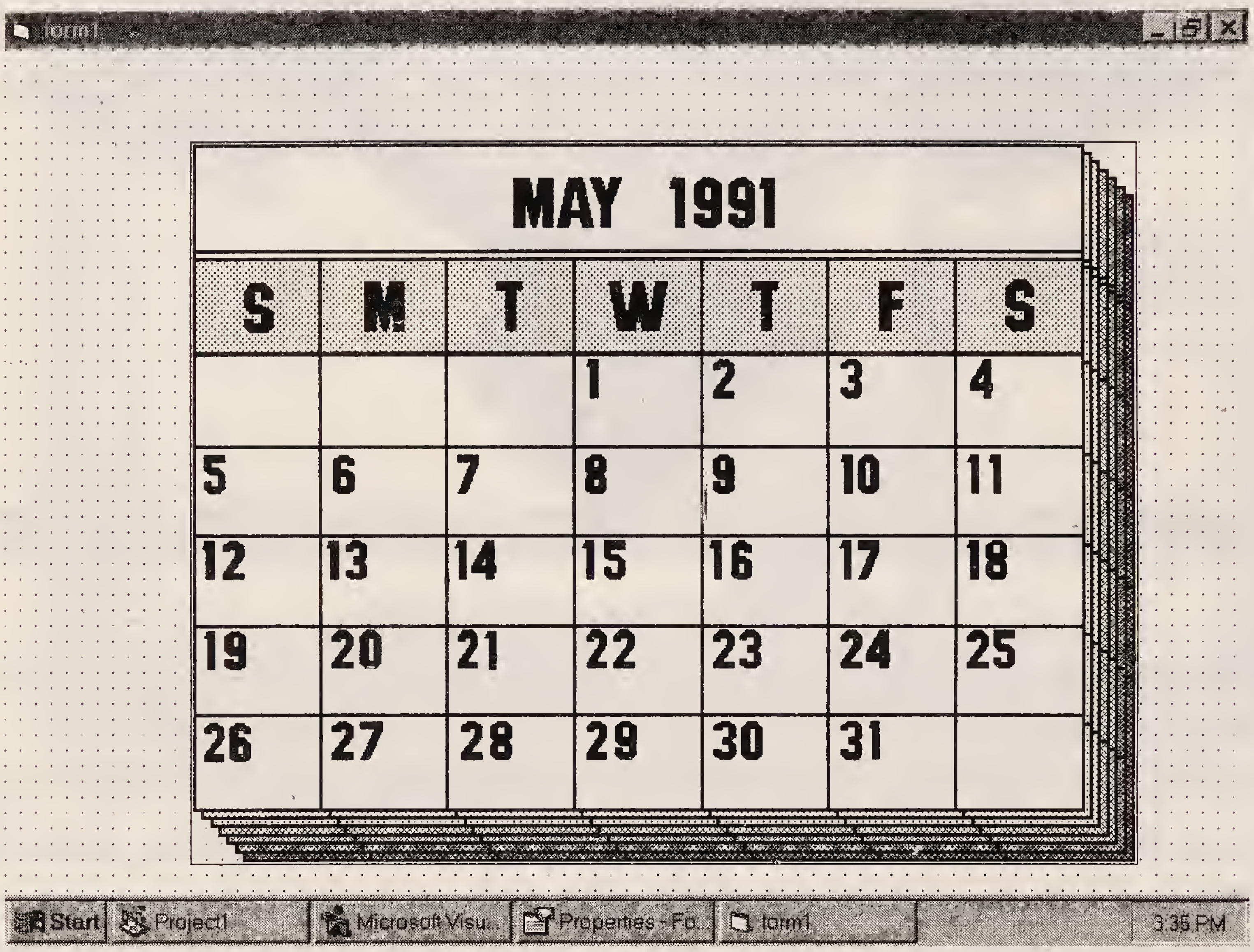
（3）在工具框中选择 Image（图像框）控件，在 Form1窗口中画出一矩形框体。

（4）选择图像框的 Picture 属性，在打开的 Load Picture 对话框中选择 bitmap、icon 或 metafile 中的一种图形，然后单击 OK 按钮。

也可以画好图像框后，打开 Code 窗口。选择 Form1_ Load 事件过程，在过程头与过程尾中插入代码：

```
Privcate Sub Form1_ Load()
        Image1. Picture=LoadPicture(FileToOpen)
End Sub
```

（5）在第四步的第一种方法中，单击 OK 按钮后，即马上在图像框中显现出所选择的图形。而第二种方法，刚开始在图像框中并没有图形，在单击 Run 菜单中的 Start 命令，或单击工具栏中的 Start 命令后，在后来出现的 Form 窗口中的图像框中出现所选中的图形，如图9.8所示。



MAY 1991

S	M	T	W	T	F	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

图9.8　运行程序后在图像框中出现图形

图像框不像图片框一样，它没有 Autosize 属性。但它一开始就可以自动改变大小以适应图形的需要，来显示一个完整的图形。这样如果一个位图(bitmap)比较大，在 Form 窗口比较小的情况下，图像框的边界也会被 Form 窗口的边界所截断。这就需要编辑一段代码使得图像框适应 Form 窗口，随着 Form 窗口的大小而改变。但当我们查看图像框的属性时却发现，没有 Align 属性。这样我们就得利用别的方法，但图像框的属性中我们看到它有 Top(高)、Left(左边)、Height(高度)、Width(宽度)属性，当这些属性值被改变时，我们会看到图像框的形状及大小也发生了改变。下面我们就利用这些属性设计一段代码，使图像框的大小适应 Form 窗口的大小。

（1）选择 File 菜单中的 New Project 命令。

（2）利用工具框中的 Image(图像框)控件在 Form 窗口内随意画一矩形框。

（3）选择 Properties 窗口中的 Top、Left 属性，分别将它们设置为0，则图像框定位在 Form 窗口的左上角，如图9.9所示。

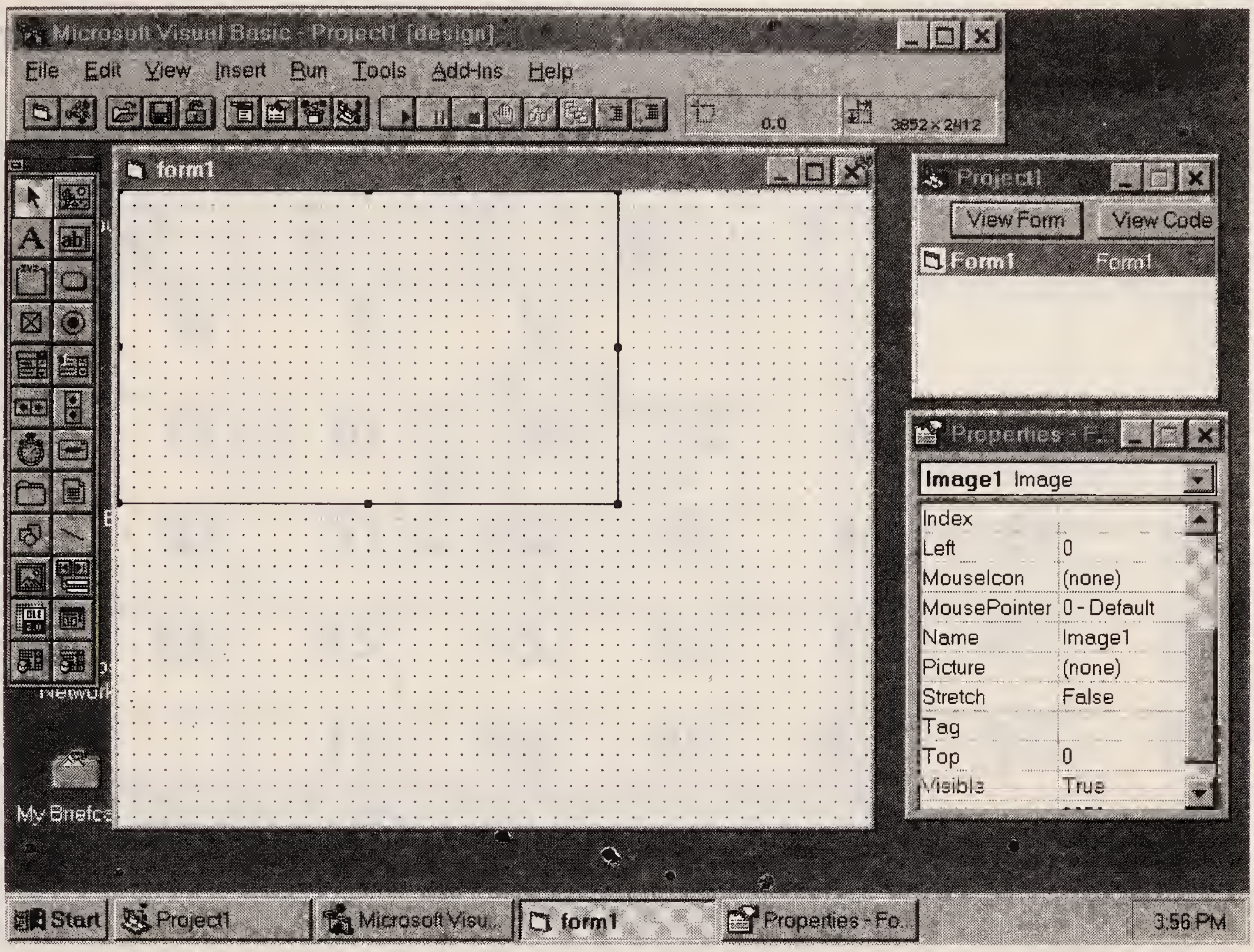

Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
0,0
3852 x 2412
form1
Project1
View Form
View Code
Form1 Form1
Properties - F...
Network
My Briefca
Start
Project1
Microsoft Visu...
form1
Properties - Fo...
3:56 PM

Image1 Image

Index
Left 0
MouseIcon (none)
MousePointer 0 - Default
Name Image1
Picture (none)
Stretch False
Tag
Top 0
Visible True

图9.9　选择 Top 为0，Left 为0后的图像框

（4）然后打开 Code 窗口，输入以下代码：

```
Private Sub Form_Load()
        Image1.Picture=Loadpicture("C:\Vb\metafile\business\calendar.wmf")
        Image1.Height=Form1.ScaleHeight
        Image1.Width=Form1.ScaleWidth
End Sub
Private Sub Form_Resize()
        Image1.Height=Form1.ScaleHeight
        Image1.Width=Form1.ScaleWidth
End Sub
```

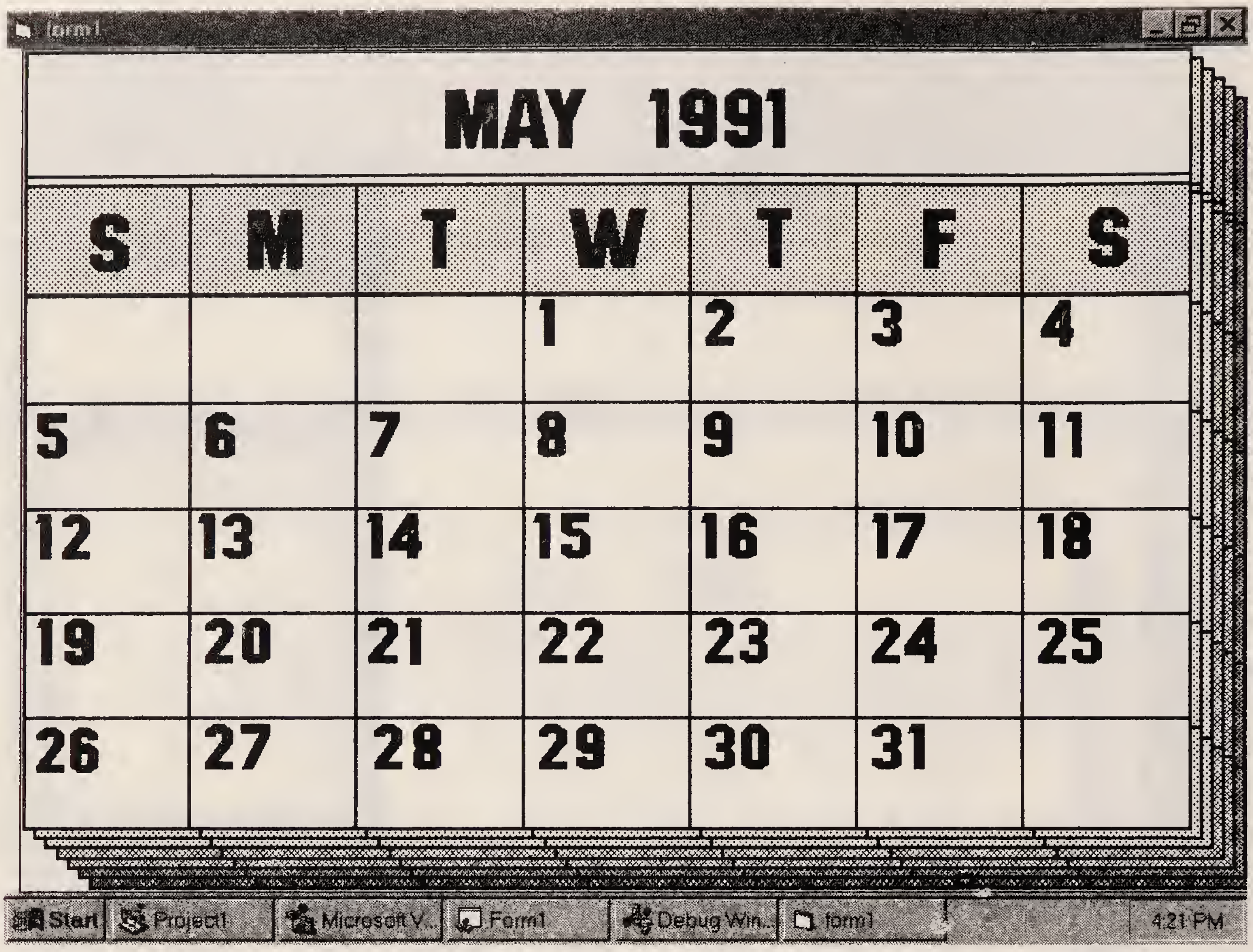
这段代码就是首先使图像框的宽度与高度和 Form 窗口的相同，在 Form 窗口的宽、高发生改变时，图像框的高、宽度也相应发生改变。

（5）运行程序。选择 Run 菜单中的 Start 命令或单击工具栏中的 Start 按钮。

程序运行后，我们发现图像框与 Form 窗口相重合，且随着 Form 窗口的改变而改变大小，如图9.10所示。

但在运行程序时，我们会发现，每当改变窗体窗口的大小时，图像框都将重画。先画出将图像框的高度与 Form 窗口对齐的过程，然后画出图像框的宽度与 Form 窗口宽度对齐的过程。如果想把这些重画过程隐去，可以对 Form_Resize 事件过程修改代码如下。

```
Private Sub Form_Resize()
        Image1.Visible=False
        Image1.Height=Form1.ScaleHeight
        Image2.Width=Form2.SadleWidth
```



MAY 1991

S	M	T	W	T	F	S
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

图9.10　图像框改变大小与 Form 窗口相匹配

```
        Image1.Visible=True
    End Sub
```

再运行程序时，将图像框的高度与 Form 窗口高度对齐这一过程将不再画出。

在前面的图像框工具中，当框体中的图形为位图或图标时，这些图形不能自动改变形状来适应框体的大小。而在图像框中，我们可以利用它的 Stretch 属性来拉伸位图或图标，使其适应图像框的大小。

(1) 首先选择 New Project 命令。

(2) 利用工具框中的 Image(图像框)工具在 Form 窗口上画一矩形框体，选择其 Properties 窗口中的 Stretch 属性，将其设置为 True。

(3) 选择 Picture 属性，在出现的 Load Picture 对话框中选择一图标，单击 OK 按钮，则马上在图像框中出现了拉伸后的一图标图形。如图9.11所示。

当然这一过程也可用代码表示。在 Form 窗口中画完图像框控件后，打开 Code 窗口，在 Form_Load()过程中输入以下代码：

```
    Private Sub Form_Load()
        Image1.Stretch=True
        Image1.Picture=LoadPicture("C:\Vb\icons\elements\snow.ico")
    End Sub
```

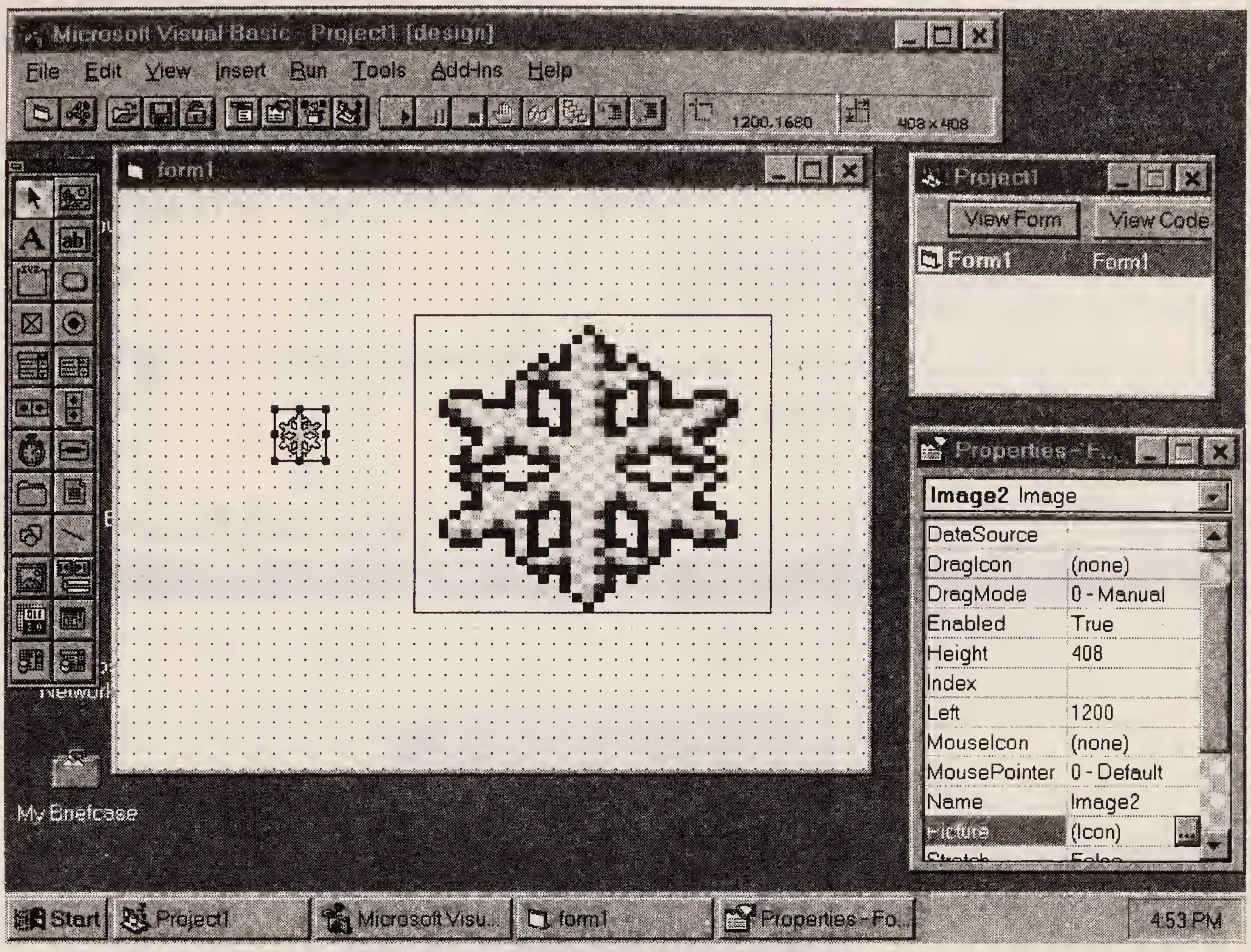

Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
1200,1680
408 x 408
form1
Project1
View Form
View Code
Form1 Form1
Properties - F...
My Briefcase
Start
Project1
Microsoft Visu...
form1
Properties - Fo...
4:53 PM

Image2 Image

DataSource	
DragIcon	(none)
DragMode	0 - Manual
Enabled	True
Height	408
Index	
Left	1200
MouseIcon	(none)
MousePointer	0 - Default
Name	Image2
Picture	(Icon)

图9.11　拉抻后与拉伸前的 Snow 图标

如果想让图标随着 Form 窗口的大小改变而改变，则按照前面讲的将图像框的大小适应 Form 窗口大小的方法设计代码即可。

在图9.11我们看到拉伸后的图标确实变大了。不过我们看到拉伸后的图标比拉抻前要模糊的多。所以一般我们并不用 Stretch 属性，只有对图形进行编辑时才使用这一属性。

9.1.3　保存图像内容

当用户为图片框或图像框绘制或装入了图形后，为了保存这一图形，就要用 SavePicture 命令把图形保存到磁盘上。这一命令格式为：

Save Picture 对象名.属性，文件名

文件名为指定的带有扩展名 BMP（位图）、ICO（图标）或 WMF（元文件）的文件。对象名指选定的图片框或图像框的控制名，如 Picture1 或 Image1，属性为图片框或图像框中的图形的属性，如 Picture 属性或 Image 属性。关于 Picture 属性在前面的例题中我们已很熟悉了。图片框的属性为 Picture，图片以控件中的当前格式保存，可以是位图文件、图标文件或元文件。

而 Image 属性常设置成该控件的图形内容。这个属性通过使用方法（method）把图形加至该控件而完成间接设置。如果图片框属性设置为 Image，则不管给定的文件名及其扩展名是什么，图像文件都被加上 BMP 扩展名，作为一个 Windows 位图文件保存。

下面，我们就设置一个图像框，然后将图像框中的内容保存起来。

(1) 首先，选择 File 菜单中的 New Project 命令。

(2) 选择工具框中的图像框工具在 Form 窗口内画一矩形框体。

(3) 打开 Code 窗口，选择 Form _ Click 事件，并在 Form _ Click 事件过程头与过程尾之间输入下列代码：

```
Private Sub Form _ Click()
        Image1.Picture=LoadPicture(″C:\Vb\metafile\business\computer.wmf″)
        SavePicture Image1.Picture,″Computer1.BMP″
End Sub
```

(4) 运行程序，然后在运行时的 Form 窗口上单击鼠标左钮。其屏幕如图9.12所示。

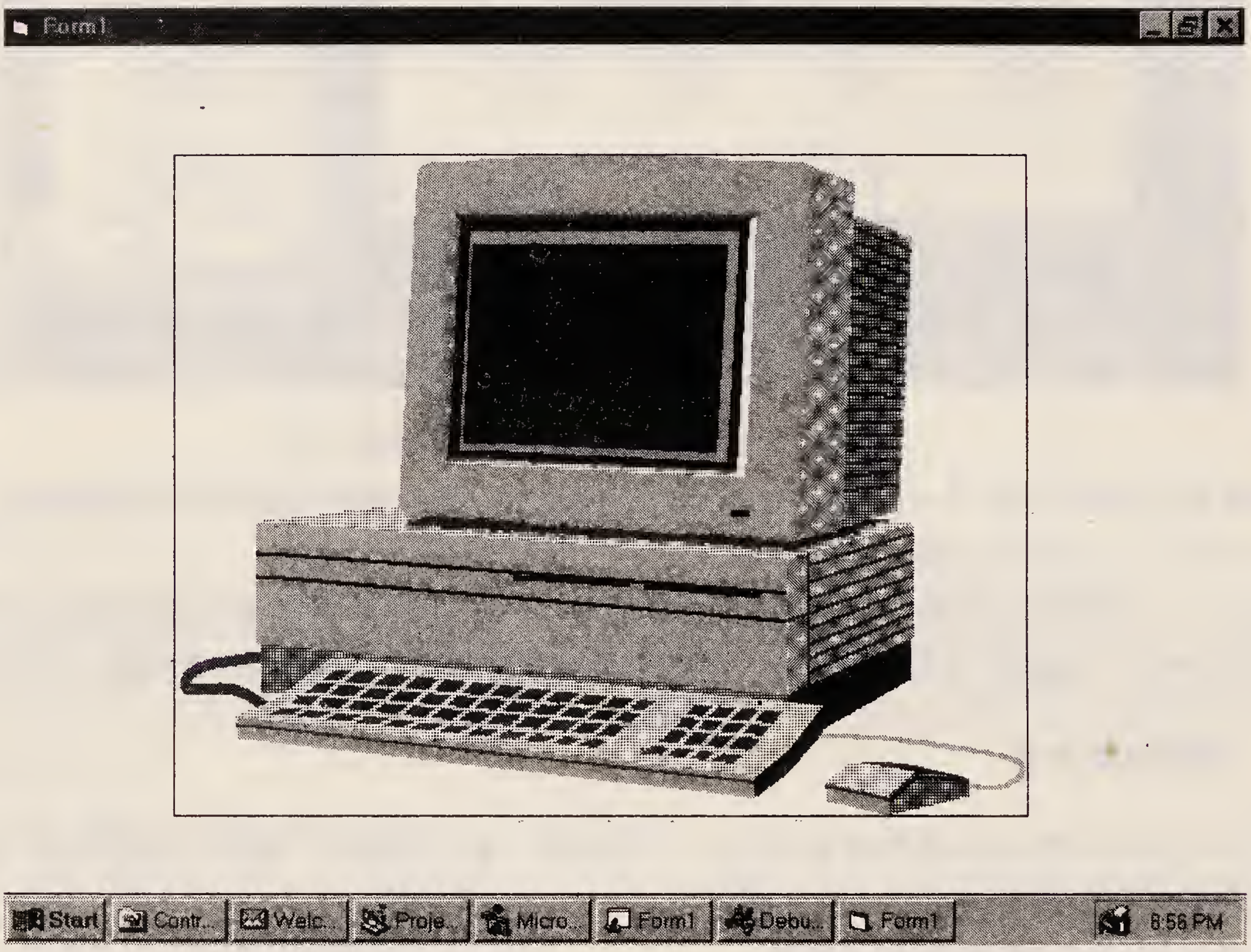
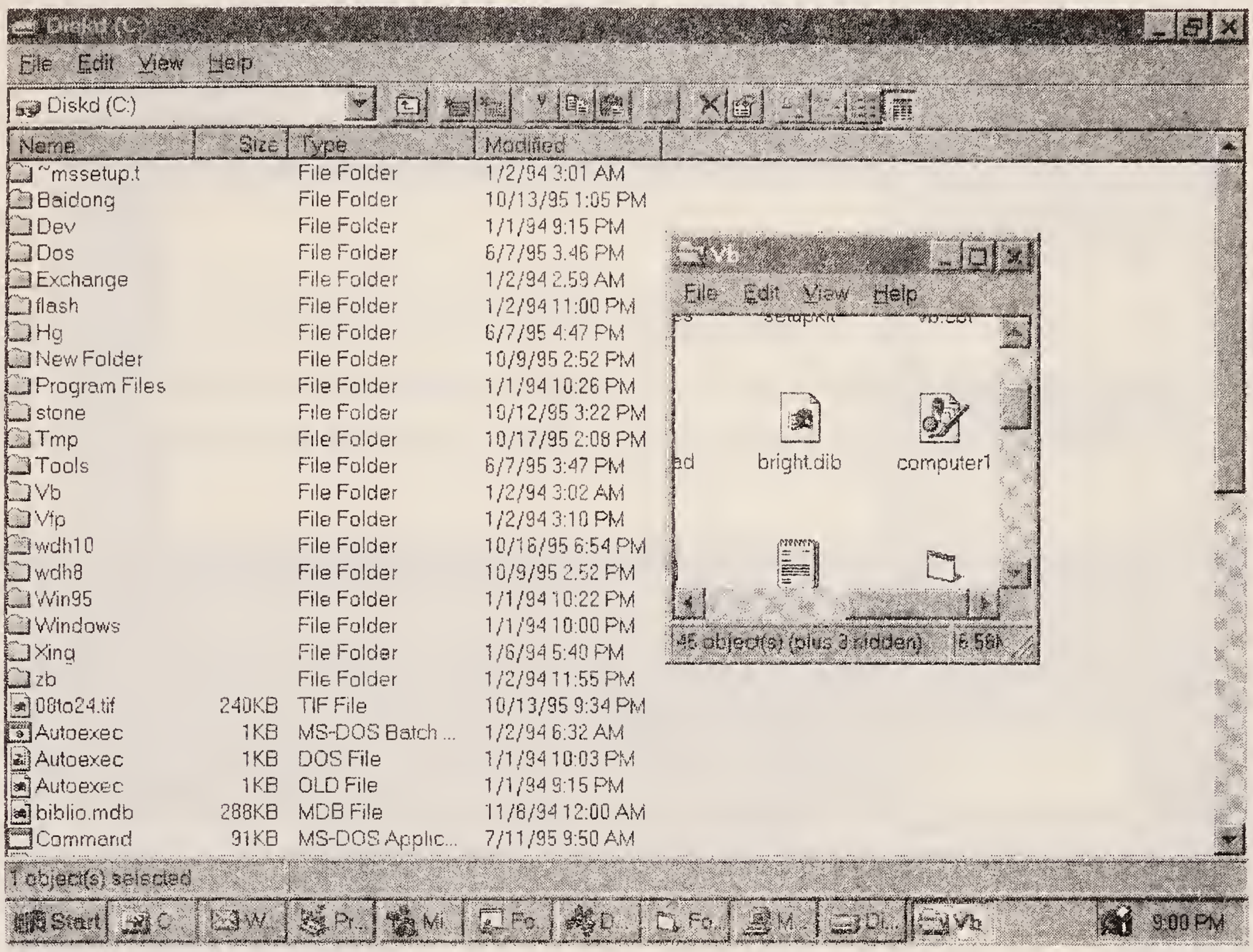


图9.12 图像框中的 Computer 图形

这时选择 Properties 窗口的 Picture 属性，在打开的 Load Picture 对话框中，选择当前路径 C:\Vb，我们会发现在对话框左边的 File 列表框中出现了 Compter1.BMP 文件。

如果回到 Windows 95桌面系统，打开 My Computer 窗口，双击 MS-dos _6[C:]图标，进入 C 盘窗口，再打开 Vb 文件夹。在 Vb 文件夹中我们会发现多了一个 computer1图标，如图9.13所示。

一般来说，用户自己画的一些图形可以用 SavePicture 函数将其保存起来，以免丢失。



Name	Size	Type	Modified
~mssetup.t		File Folder	1/2/94 3:01 AM
Baidong		File Folder	10/13/95 1:05 PM
Dev		File Folder	1/1/94 9:15 PM
Dos		File Folder	6/7/95 3.46 PM
Exchange		File Folder	1/2/94 2.59 AM
flash		File Folder	1/2/94 11:00 PM
Hg		File Folder	6/7/95 4:47 PM
New Folder		File Folder	10/9/95 2:52 PM
Program Files		File Folder	1/1/94 10:26 PM
stone		File Folder	10/12/95 3:22 PM
Tmp		File Folder	10/17/95 2:08 PM
Tools		File Folder	6/7/95 3:47 PM
Vb		File Folder	1/2/94 3:02 AM
Vfp		File Folder	1/2/94 3:10 PM
wdh10		File Folder	10/16/95 6:54 PM
wdh8		File Folder	10/9/95 2.52 PM
Win95		File Folder	1/1/94 10:22 PM
Windows		File Folder	1/1/94 10:00 PM
Xing		File Folder	1/6/94 5:40 PM
zb		File Folder	1/2/94 11:55 PM
08to24.tif	240KB	TIF File	10/13/95 9:34 PM
Autoexec	1KB	MS-DOS Batch ...	1/2/94 6:32 AM
Autoexec	1KB	DOS File	1/1/94 10:03 PM
Autoexec	1KB	OLD File	1/1/94 9:15 PM
biblio.mdb	288KB	MDB File	11/8/94 12:00 AM
Command	91KB	MS-DOS Applic...	7/11/95 9:50 AM

1 object(s) selected

图9. 13　Vb 文件夹中的 Computer1图标

9. 1. 4　图片框与图像框的区别

图片框与图像框都可以用来显示 bmp（位图）、icons（图标）或 metafile（元文件）中的图形，但它们之间也是有区别的。

（1）首先图像框比图片框占用更少的内存，描绘的更快。一般为了占用少的内存，最好使用图像框。当然，要先对这两种控件进行比较，在确实用不到图片框时，再用图像框取代图片框。

（2）图片框控件可以包含其它控件，例如，选择工具框中的图片框工具在 Form 窗口上画一框体。然后选择命令按钮控件，在图片框中画一命令按钮。

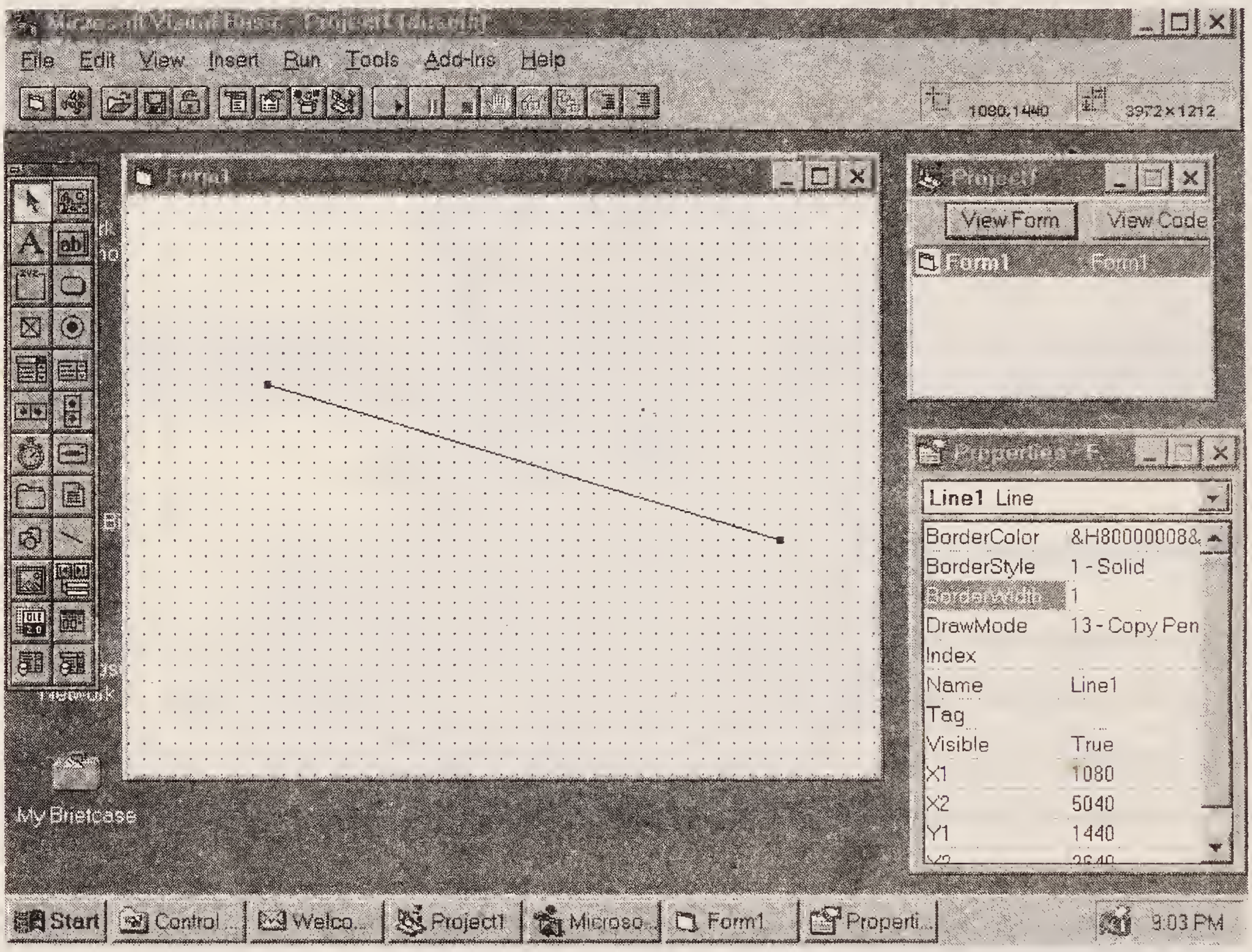
当我们移动图片框时，图片框中的命令按钮也随着一起移动，并且在图片框中相对图片框的位置保持不变。图片框中的按钮只能在图片框内移动，不能移到图片框以外。

而图像框则不能在其框体内保存其他控件。例如，选择工具框的图像框控件，在 Form 窗口内画一图像框。再选择命令按钮控件，在图像框中画一命令按钮。当移动图像框时，命令按钮并不随着移动。命令按钮也可以移到图像框以外单独存在。用户可以按着以上步骤试着操作一下。

9.2　绘图工具

9.2.1　Line(直线)工具

直线控件可以用来在 Form 上画线。画线的过程是单击工具框中的直线控件，把鼠标指针移到 Form 窗口内，这时鼠标指针变为十字线。选中直线的起点，然后将十字线移到这一点，单击鼠标，并拖动鼠标，移到直线的终点，放开鼠标。这时即在 Form 窗口中出现一直线。如图9.14所示。


File
Edit
View
Insert
Run
Tools
Add-Ins
Help
1080,1440
3972×1212
Form1
Project1
View Form
View Code
Form1　Form1
Properties - F
My Briefcase
Start
Control...
Welco...
Project1
Microso...
Form1
Properti...
9:03 PM

Line1 Line

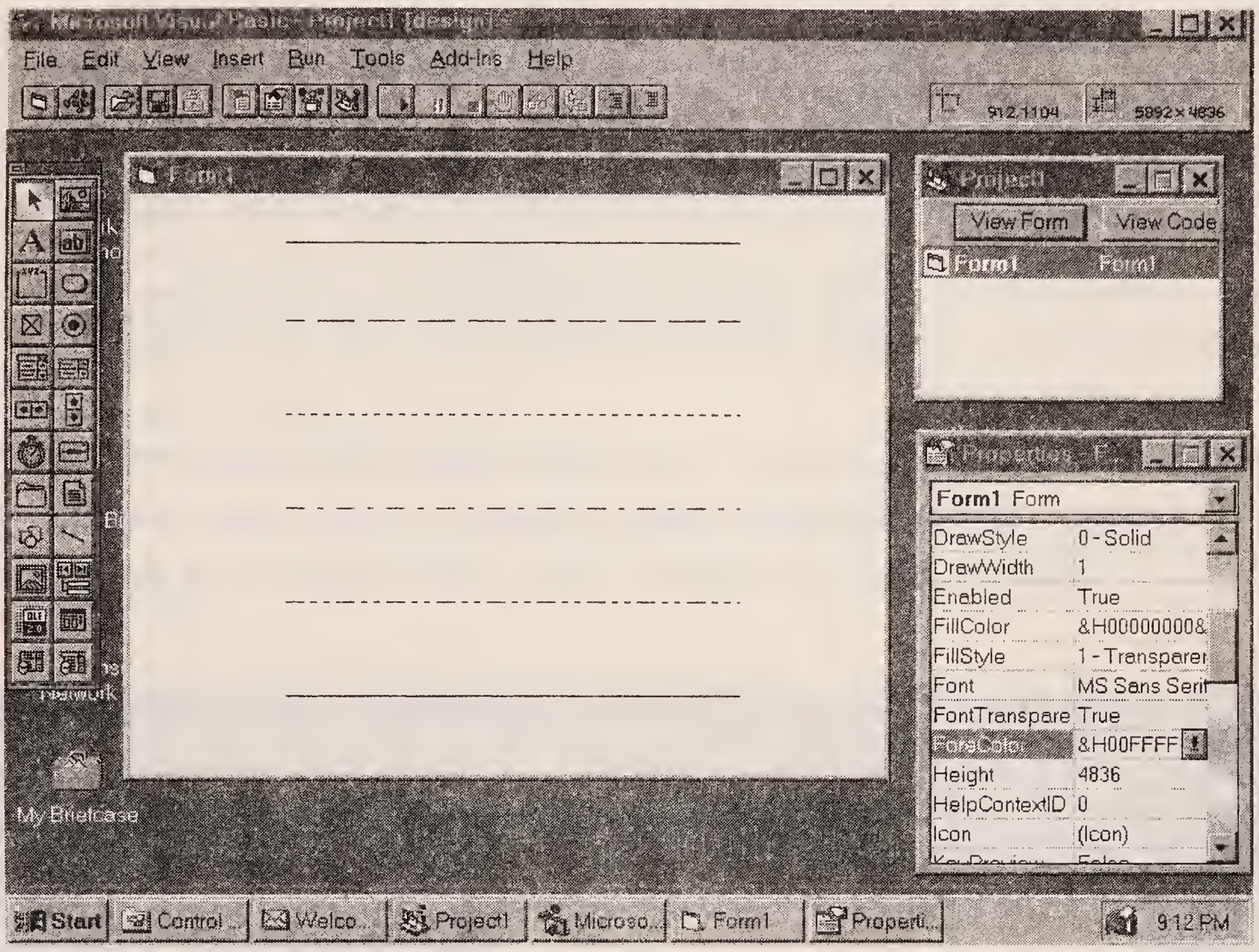
BorderColor	&H80000008&
BorderStyle	1 - Solid
BorderWidth	1
DrawMode	13 - Copy Pen
Index	
Name	Line1
Tag	
Visible	True
X1	1080
X2	5040
Y1	1440

图9.14　用直线控件在 Form 窗口中画一直线

直线控件还可以显示在图片框(Picture Box)以及框体(Frame)中。要想移动直线，可以用鼠标来移动它，也可以通过改变它的 X1、X2、Y1或 Y2属性即重设 X1、X2、Y1或 Y2值来实现。一个直线对象的属性 X1、X2、Y1和 Y2表示了窗体中在当前度量系统下直线顶点的几何坐标，即(X1,Y1),(X2,Y2)表示了笛卡尔坐标系上一条直线的起点和终点。

在图9.14中我们看到画出的直线为实心线，这是由直线的 BorderStyle 属性决定的。直线的 BorderStyle 属性决定了直线的长划线方式，其表示的标记式样如图9.15所示。

0　透明线
1——实心线
2-- — — — — — —长划线
3......由点组成的直线
4—.—.—.—点—划线
5—..—长划—点—点—长划线
6——内部实线

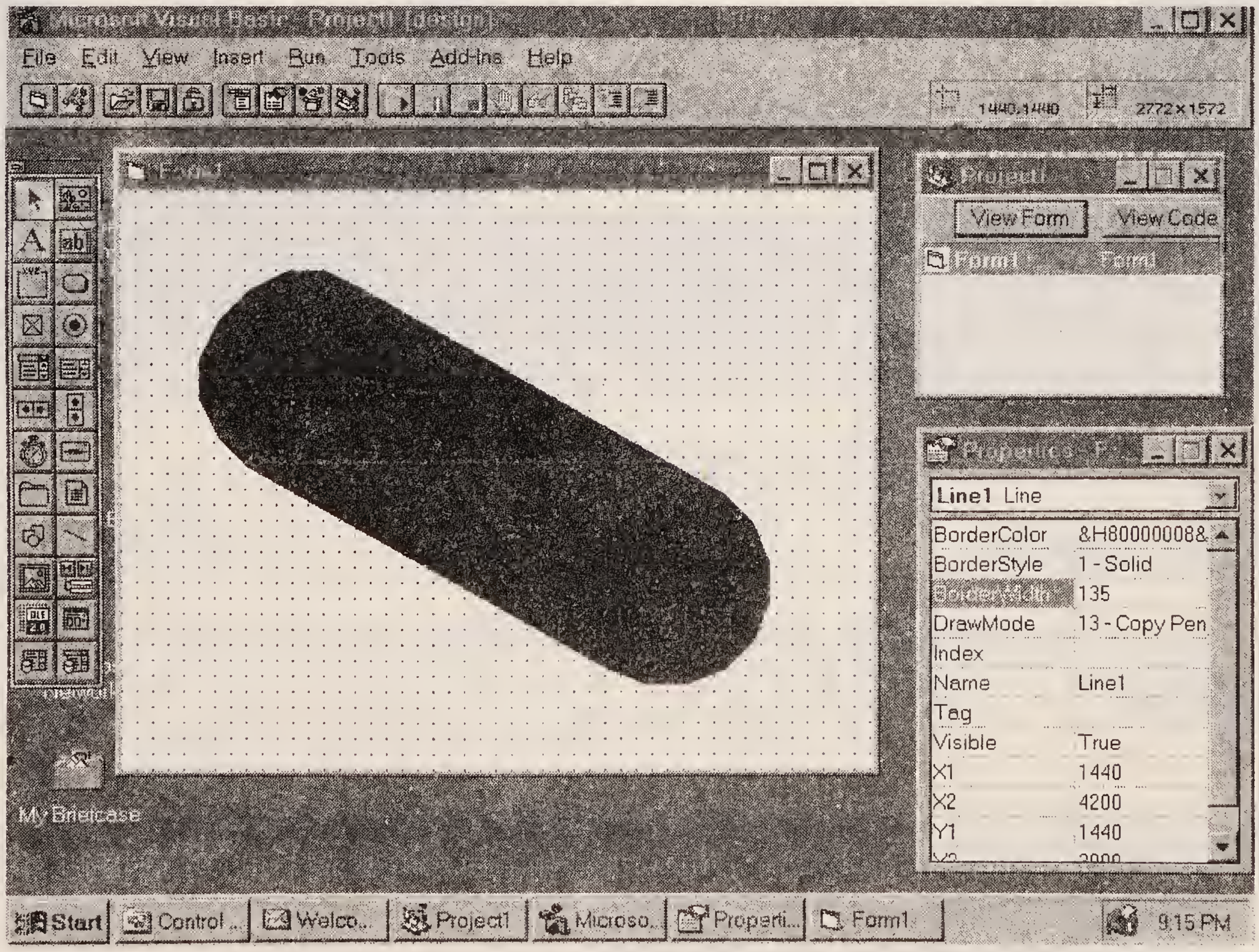

Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
912,1104
5892×4836
Form1
Project1
View Form
View Code
Form1 Form1
Properties - F...
My Briefcase
Network
Start
Control...
Welco...
Project1
Microso...
Form1
Properti...
9:12 PM

Form1 Form	
DrawStyle	0 - Solid
DrawWidth	1
Enabled	True
FillColor	&H00000000&
FillStyle	1 - Transparer
Font	MS Sans Serif
FontTranspare	True
ForeColor	&H00FFFF
Height	4836
HelpContextID	0
Icon	(Icon)

图9.15　BorderStyle 不同值画出不同的线

BorderStyle 的缺省值为1，即实心线。

解释器认为直线控件的标记是由笔尖画出的。笔尖宽度由 BorderWidth 属性决定。BorderWidth 属性设置可从1到8192。随着属性值的增大，直线变得越来越粗。如果属性值太大，这时直线控件画出的就不像一条直线了，而是变成黑糊糊的一团。如图9.16所示。


Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
1440,1440
2772×1572
Form1
Project1
View Form
View Code
Form1 Form1
Properties - F...
My Briefcase
Network
Start
Control...
Welco...
Project1
Microso...
Properti...
Form1
9:15 PM

Line1 Line	
BorderColor	&H80000008&
BorderStyle	1 - Solid
BorderWidth	135
DrawMode	13 - Copy Pen
Index	
Name	Line1
Tag	
Visible	True
X1	1440
X2	4200
Y1	1440

图9.16　BorderWidth 值过大而使直线变形

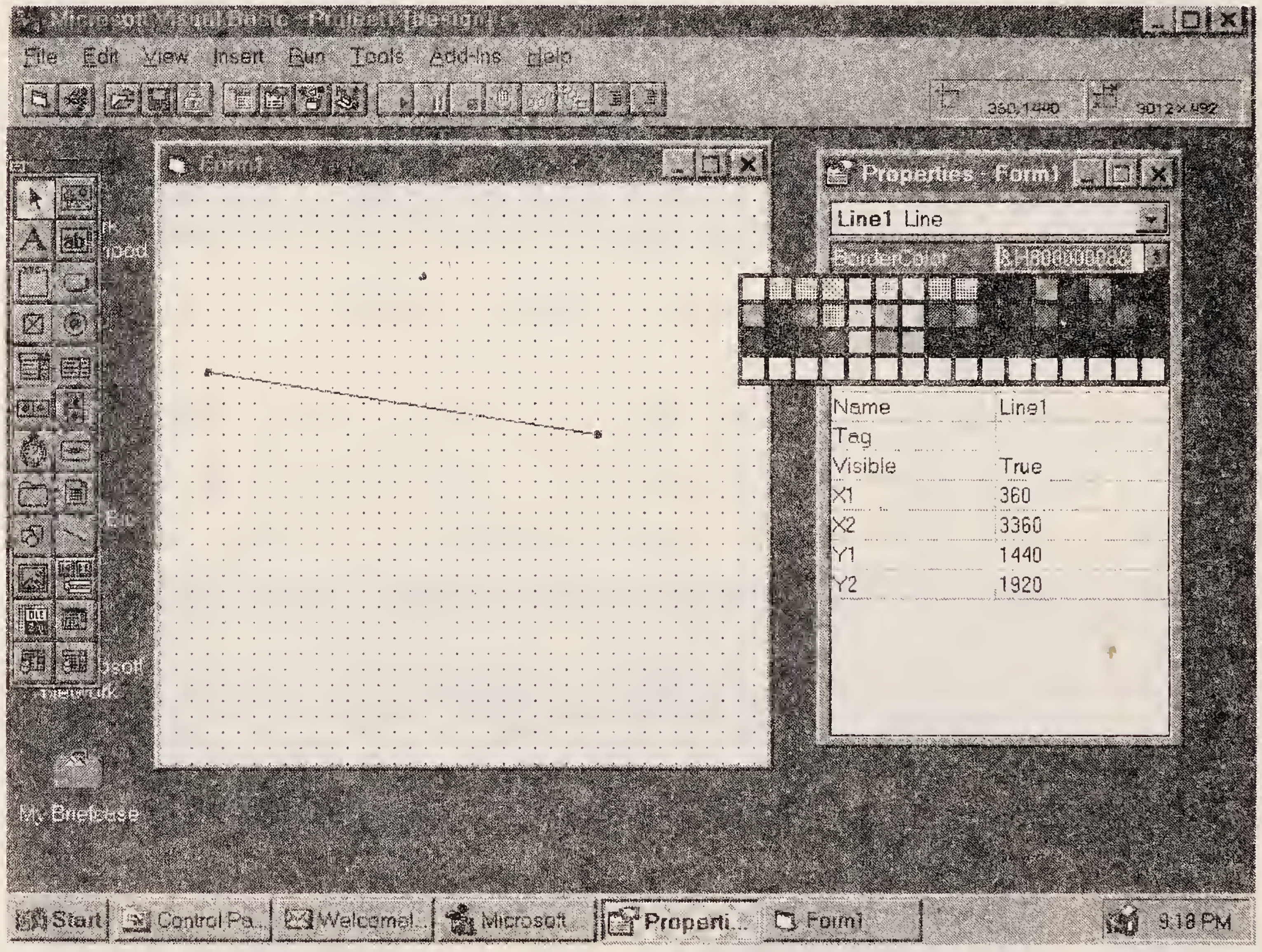
BorderStyle 属性值的设置将对 BorderWidth 属性产生影响。

BorderStyle	对 BorderWidth 的影响
0	BorderWidth 的设置被忽略
1—5	边界宽度从边界中心开始计算。控件的宽和高从边界的中心开始测量
6	边界宽度从边界外沿向内计算。控件的宽和高从边界外沿开始测量

若 BorderWidth 大于1，BorderStyle 的有效设置是1（实心线）和6（内部实线）。

BorderWidth 的缺省值为1。

直线的颜色还可以通过 BorderColor 属性来进行设置。单击 Properties 窗口中的 BorderColor 属性，在其右边的设置框中会出现一带箭头的按钮。单击这一按钮，即出现一颜色对话框，如图9.17所示。用鼠标单击对话框中的某一颜色，在 Form 窗口中，直线即马上变为新选的颜色。



Microsoft Visual Basic - Project1 [design]
File
Edit
View
Insert
Run
Tools
Add-Ins
Help
360,1440
3012 x 492
Form1
Properties - Form1
BorderColor
My Briefcase
Start
Control Pa...
Welcome!...
Microsoft...
Properti...
Form1
9:18 PM

Line1 Line

Name	Line1
Tag	
Visible	True
X1	360
X2	3360
Y1	1440
Y2	1920

图9.17　**BorderColor 颜色对话框**

9.2.2　用 Line 方法绘图

除了用 Line 控件直接在窗体或图片框等内部画直线外，还可以用 Line 方法在一个对象上画直线和矩形。

格式为：

[对象名.]Line[[Step](x1,y1)]-[step](x2,y2)[,[color][,B[F]]]

部分	描述
对象名	画出直线或矩形的对象（窗体，图片框等）
Step	确定坐标起始点的关键点相对于当前图像位置，由 CX 和 CY 属性给定

（续表）

部分	描述
X1,Y1	单精度值指定直线或矩形起点坐标。对象的刻度特性(ScaleMode,ScaleLeft 等)确定刻度单位。如果省略了,就由 CX 和 CY 来确定直线的起点
Step	确定相对于直线起点的终点坐标
X2,Y2	确定要画出的直线终点坐标的单精度数值。这里不能省略
Color	画直线时标定 RGB 颜色的长整数。如果省略,就使用 ForeColor 属性。可以使用 RGB 函数或 QBColor 函数来确定颜色
B	通过指定框的对角座标来画一个框
F	如果使用了选项 B,选项 F 就表示由与画框相同的颜色来填充框。不能在不用 B 的情况下使用 F。如果仅用 B 而不用 F,就用当前 FillColor 和 FillStyle 填充框。Fillstyle 的缺省值是透明的

下面我们将介绍几个 Line 方法用到的属性:

1. CurrentX,CurrentY 属性

适用于 Form、图片框和 Printer 对象。确定下一次打印或绘画的水平(CurrentX)和垂直(CurrentY)坐标,设计时不可用。

坐标是从对象的左上角开始计算的。对象左边界处的 CurrentX 为0,对象顶边界处的 CurrentY 为0。坐标在缺省情况下以像素为单位表示,或根据当前用 ScaleHeight、ScaleWidth、ScaleLeft、ScaleTop 以及 ScaleMode 属性设置的刻度表示。

在使用下面的图像方法后,CurrentX 与 CurrentY 作相应改变:

方法	把 CurrentX,CurrentY 设置成
Circle	对象的中心
Cls	0,0
EndDoc	0,0
Line	直线的终止点
NewPage	0,0
Print	下一个打印位置
Pset	画点所在位置

2. ScaleMode 属性

适用于 Form、图片框和 Printer 对象。在使用绘图方法或定位控件时,确定对象坐标的度量单位。它的属性设置如下:

设置值	描述
0	表示 ScaleHeight、ScaleWidth、ScaleLeft 和 ScaleTop 属性中的一个或几个设置为常规值
1	(缺省值)点素(每一逻辑英寸1440点素,每一逻辑厘米567点素。)
2	点(每逻辑英寸72点)
3	像素(监视器和打印设置分辨率最小单位)
4	字符(水平的=120点素/单位,垂直的=240点素/单位)
5	英寸
6	毫米
7	厘米

ScaleMode 属性与 ScaleHeight、ScaleWidth、ScaleLeft 和 ScaleTop 属性相互作用。

当后四种属性设置成任意值，可自动将 ScaleMode 设置为0。

将 ScaleMode 设置为一个大于0的数，会改变 ScaleHeight 和 ScaleWidth 成为新的度量单位，并且置 ScaleLeft 和 ScaleTop 为0。CurrentX 和 CurrentY 属性的设置值将变化并反映到当前点的新坐标上。

3. DrawMode 属性

适用于 Form、Line、图片框、Printer 图像和形状。决定图形方法的输出或者外形或直线控件的外观。

DrawMode 属性设置如下：

设置值	描述
1	输出为黑色(Blackness)
2	输出为 MergePen(设置为15)的非(Not Merge Pen)
3	把单背景色和笔颜色的非进行与后输出(Mask Not Pen)
4	输出为 Copy Pen(设置为13)的非(Not Copy Pen)
5	把笔颜色与显示色的非进行与后输出(Mask Pen Not)
6	输出为显示颜色的非(Invert)
7	把笔和显示颜色异或后输出(Xor Pen)
8	输出为 Mask Pen(设置为9)颜色的非(Not Mask Pen)
9	把笔颜色和显示颜色与后输出(Mask Pen)
10	输出为 Xor Pen(设置为7)颜色的非(Not Xor Pen)
11	不做操作——输出保持不变。实际效果是此设置关闭绘画(Nop)
12	把显示色和笔颜色的非合并后输出(Merge Not Pen)
13	颜色由 Fore Color 属性值指定(为缺省值)(Copy Pen)
14	把笔颜色和显示色的非合并后输出(Merge Pen Not)
15	组合笔颜色和显示色(Merge Pen)
16	白色(Whiteness)

使用此特性对形状和直线控件或用图形方法绘制的图形产生直接效果。Visual Basic 将绘画图案中的第一个像素与存在于背景中相应的像素进行比较，然后施加位操作。例如，将 DrawMode 设置为7，使用 Xor 运算将绘画图案像素与背景像素进行组合。

4. DrawStyle 属性

适用于 Form、图片框和 Printer 对象。确定图形方法输出的线型。

DrawStyle 特性设置如下：

设置值	描述
0	(为缺省值)实心线
1	虚线
2	点线
3	虚线、点线
4	虚线、点一点线
5	透明
6	内部实线

若 DrawWidth 值设置大于1，则从1到4的 DrawStyle 产生一条实心线（即 DrawStyle 属性值没有改变）。若 DrawWidth 设置为1，则 DrawStyle 产生上述描述所对应的效果。

下面，举一个例子，在窗体中画7条线，每条线有不同的 DrawStyle 值。

选择 File 菜单中的 New Project 命令后，在 Code 窗口中输入下列代码：

```
Private Sub Form _ Click()
  Dim I
  ScaleHeight =8
  For I=0 To 6
      DrawStyle = I
      Line (0,I+1)-(ScaleWidth,I+1)
  Next I
End Sub
```

这段代码是设置窗体高度为8，然后在窗体上画出7条平行线。输入完代码后，运行程序。单击 Form 窗口，我们会看到在 Form 窗口上画出7条平行线，由于 DrawStyle 属性不同，我们看到这7条直线不同。并且由于一条直线的 DrawStyle 属性为5（透明线），所以只能看到6条直线。如图9.18所示。

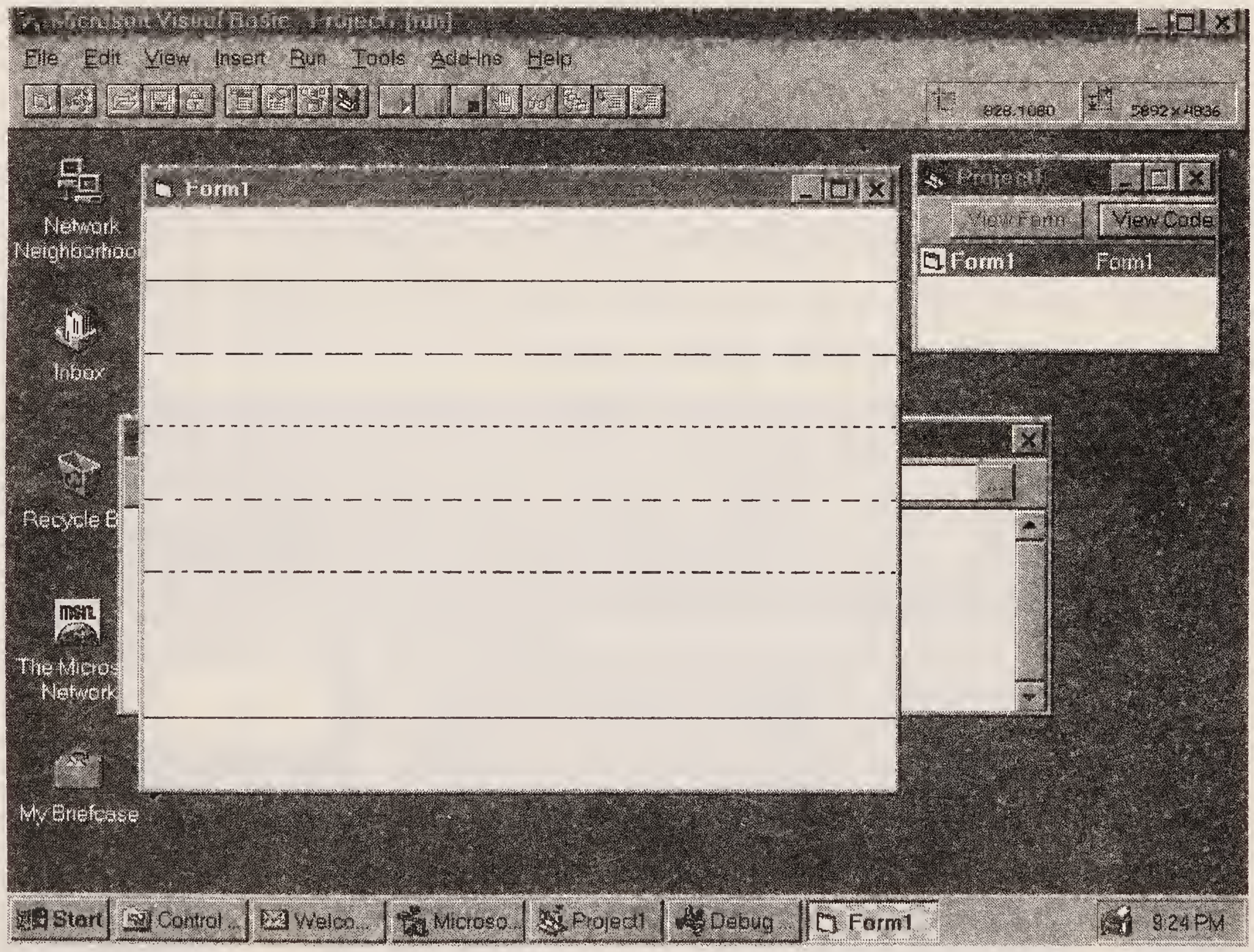

Microsoft Visual Basic - Project1 [run]
File
Edit
View
Insert
Run
Tools
Add-Ins
Help
928,1080
5892x4836
Form1
Project1
View Form
View Code
Form1
Form1
Network Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Start
Control...
Welco...
Microso...
Project1
Debug...
Form1
9:24 PM

图9.18　不同的 DrawStyle 属性画出的直线不同

5. DrawWidth 属性

适用于 Form、图片框和 Printer 对象。确定图形的线条宽度。DrawWidth 的属性值可以从1到32767，此值用像素为单位表示线条宽度，缺省值1，即1像素宽。增加此属性值可以增加线条的宽度。

还有 ScaleHeight，ScaleWidth 属性，我们在前面的例题中已多次用到它们了。这两个属性值用来确定一个对象内部水平的（标量宽）和垂直的（标量高）单元数目。

ScaleLeft 和 ScaleTop 属性是在使用绘图方法或布局控件时，确定对象左边和顶端的水平(标量左)或垂直(标量顶)坐标，上述属性缺省值为0。

在对上述属性有了简单的认识以后，下面就举一个用 Line 方法绘图的例子。

这个例子使用 Line 方法在一个窗体上绘制一个逐渐缩小的框体。

例：首先，单击 File 菜单中的 New Project 命令，然后打开 Code 窗口，输入下列代码：

```
Private Sub Form _ Click()
    Dim CX,CY,Z,Z1,Z2,I
    ScaleMode = 3
    CX=ScaleHeight/2
    CY=ScaleWidth/2
    DrawWidth=8
    For I=50 TO 0 Step -2
      Z=I/50
      Z1=1-Z:Z2=1+Z
      ForeColor=QBColor(I Mod 15)
      Line(CX *  Z1,CY * Z1)-(CX * Z2,CY * Z2), ,BF
  Next I
End Sub
```

这段代码首先声明变量，然后设置了 ScaleMode 属性，再确定了要画的框的中心点，然后再进行计算，绘制图形，并执行循环语句，绘制逐渐变小的框体。

输完代码后，单击工具栏中的 Start 按钮，在程序运行时出现的 Form 窗口中用鼠标单击一下，即出现逐渐缩小的一些框体，如图9.19所示。

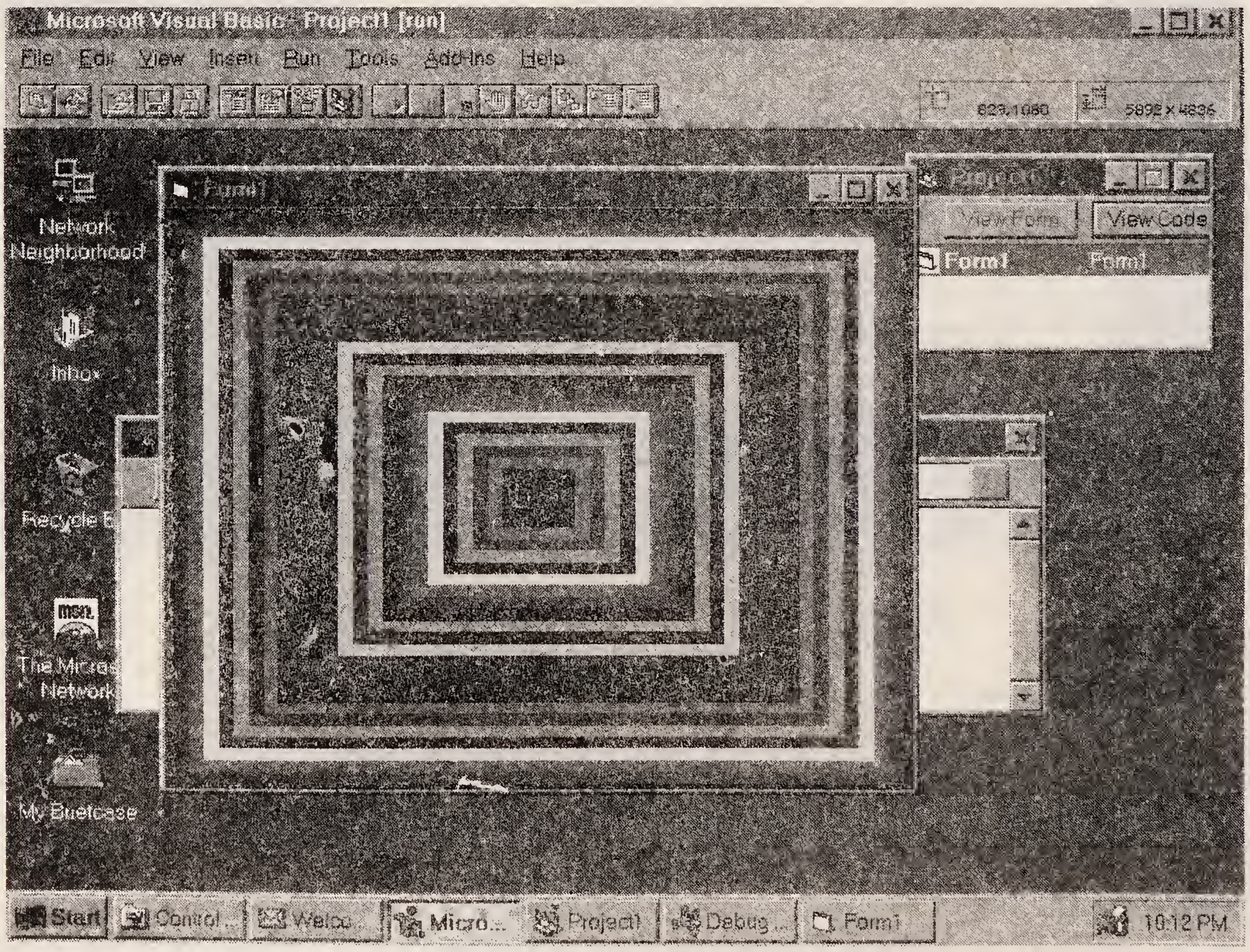

Microsoft Visual Basic - Project1 [run]
File
Edit
View
Insert
Run
Tools
Add-Ins
Help
Form1
View Form
View Code
Form1
Network Neighborhood
Inbox
Recycle B
The Micros Network
My Briefcase
Start
Control
Welco
Micro
Project1
Debug
Form1
10:12 PM

图9.19　逐渐变小的框体

9.2.3　Shape(形状)工具

形状控件是一个图形控件,可显示为矩形、正方形、椭圆、圆、圆角矩形以及圆角正方形。

在 Form 窗口中画一形状控件的过程是:

(1) 首先在工具框中选择形状控件,当鼠标指针移到 Form 窗口中时将变为“十”字。

(2) 在 Form 窗口内单击并拖动鼠标,使画出的矩形为用户所满意的形状。然后放开鼠标。

(3) 在 Properties 窗口中选择 Shape(形状)属性。在 Shape 属性的设置框中选择所需要的形状。

Shape 属性决定了 Shape 控件的形状,Shape 的属性值如下:

设置值	描述
0	(缺省值)矩形
1	正方形
2	椭圆
3	圆
4	圆角矩形
5	圆角正方形

该属性缺省时设置为0,即当用户单击 Shape 工具在 Form 窗口内绘制 Shape 图形时,刚开始都为矩形。如图9.20所示。在以上的各种情况中,控件的高度与宽度都由窗体中控件顶点的坐标来确定。属性 Shape 只影响到形状控件显示的图形元素的形状。当 Shape 为1,3,5时,它控制的形状轮廓的宽度和高度相等,形状控件显示的图形元素的实际尺寸由该控件的属性 Height 和 Width 中较小的一个设置。在 Shape 为4,5时,图形元素角度采用的圆度大小是固定的。如图9.21所示。

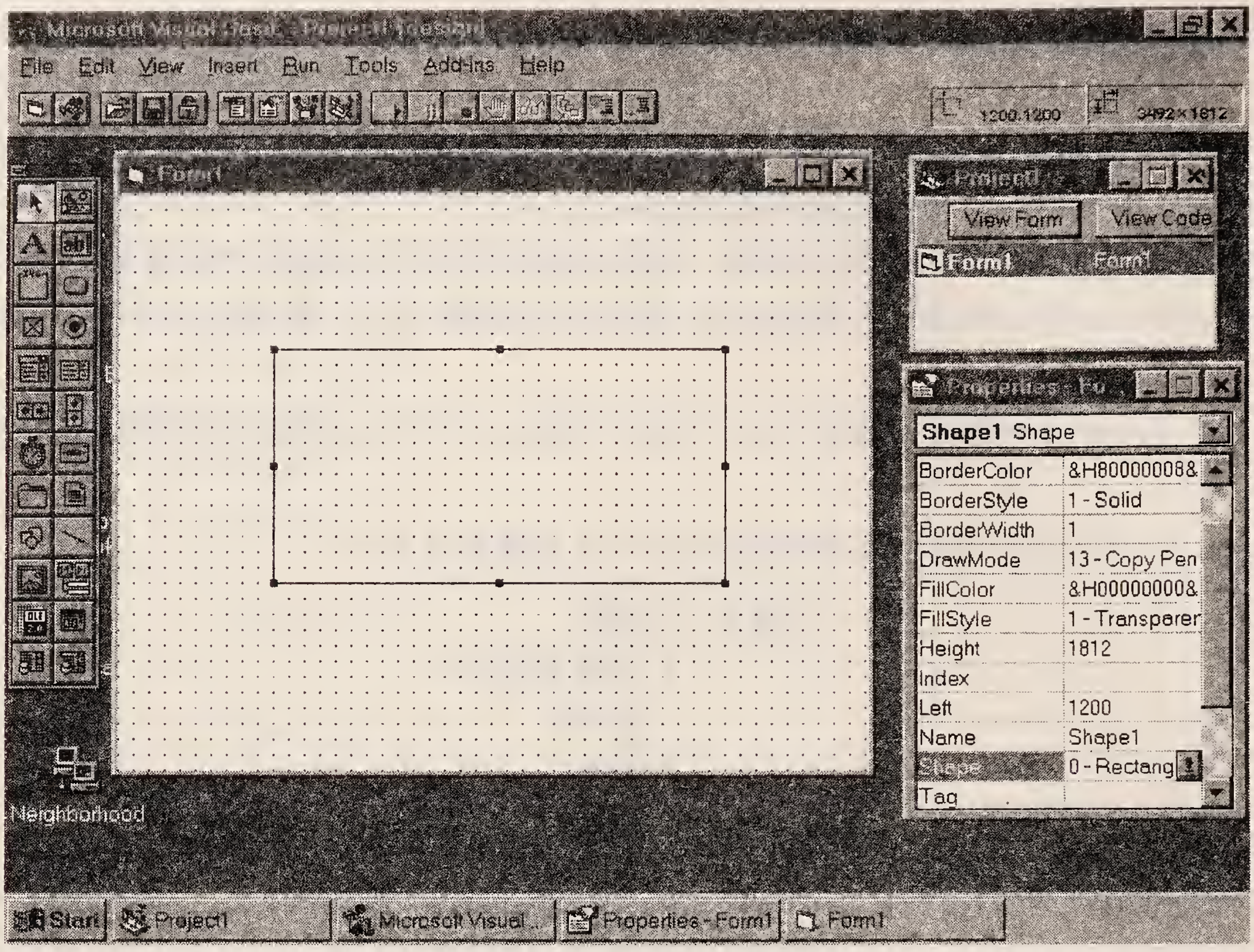
我们还可以用 FillStyle 属性为 Shape 控件进行填充。FillStyle 属性适用于 Form、Grid(网格)、图片框、Printer 对象以及 Shape 控件。

Grid(网格)的 FillStyle 属性设置如下:

设置值	描述
0	Single ——只改变活动单元的 Text 属性
1	Repeat ——改变所有被选中单元的 Text 属性

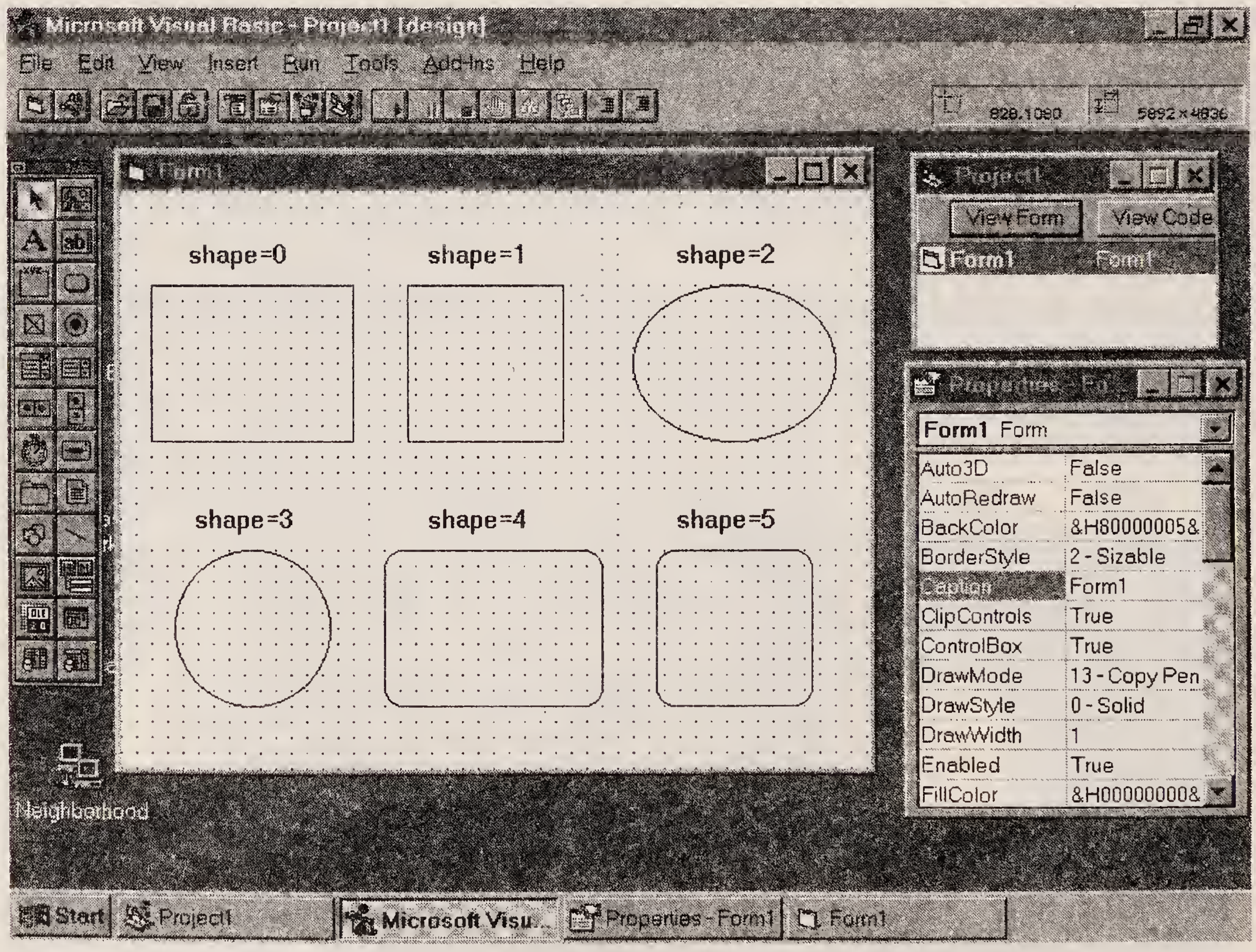
其它控件的 FillStyle 属性设置如下:

设置值	描述
0	实心
1	(缺省值)透明
2	水平线
3	垂直线
4	向上对角线
5	向下对角线
6	交叉线
7	对角交叉线


Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
1200,1200
3492×1812
Form1
Project1
View Form
View Code
Form1
Properties - Fo
Neighborhood
Start
Project1
Microsoft Visual ...
Properties - Form1
Form1

Shape1 Shape	
BorderColor	&H80000008&
BorderStyle	1 - Solid
BorderWidth	1
DrawMode	13 - Copy Pen
FillColor	&H00000000&
FillStyle	1 - Transparer
Height	1812
Index	
Left	1200
Name	Shape1
Shape	0 - Rectang
Tag	

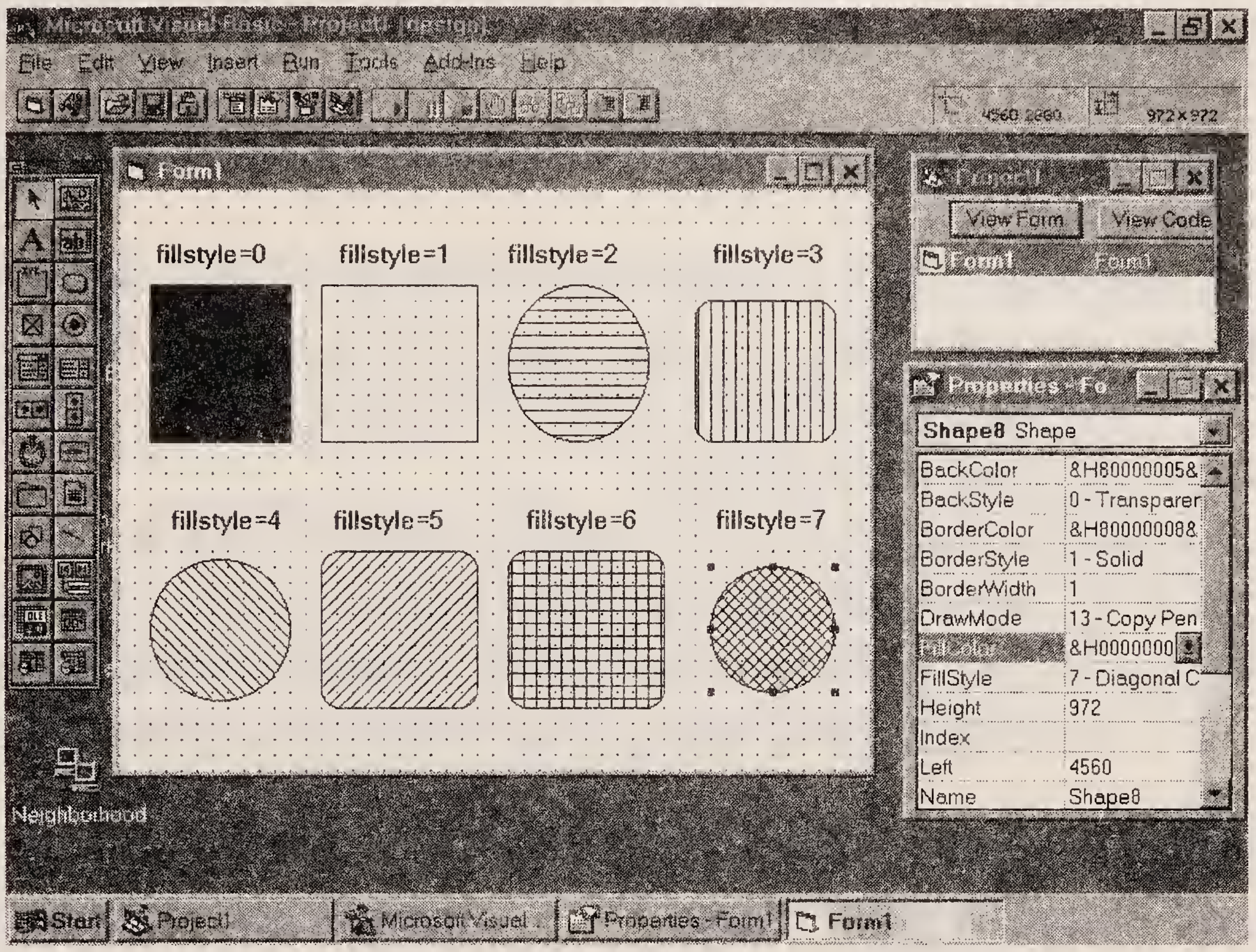
图9.20 用 Shape 控件绘图时，刚开始都为矩形


Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
828,1080
5892×4836
Form1
shape=0
shape=1
shape=2
shape=3
shape=4
shape=5
Project1
View Form
View Code
Form1
Properties - Fo
Neighborhood
Start
Project1
Microsoft Visu...
Properties - Form1
Form1

Form1 Form	
Auto3D	False
AutoRedraw	False
BackColor	&H80000005&
BorderStyle	2 - Sizable
Caption	Form1
ClipControls	True
ControlBox	True
DrawMode	13 - Copy Pen
DrawStyle	0 - Solid
DrawWidth	1
Enabled	True
FillColor	&H00000000&

图9.21 不同 Shape 属性值的 Shape 控件

对于不同的 FillStyle 属性值，Shape 控件的内部填充均不相同，如图9.22所示。


Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
4560, 2880
972 x 972
Form1
fillstyle=0
fillstyle=1
fillstyle=2
fillstyle=3
fillstyle=4
fillstyle=5
fillstyle=6
fillstyle=7
Project1
View Form
View Code
Form1
Neighborhood
Start
Project1
Microsoft Visual ...
Properties - Form1
Form1

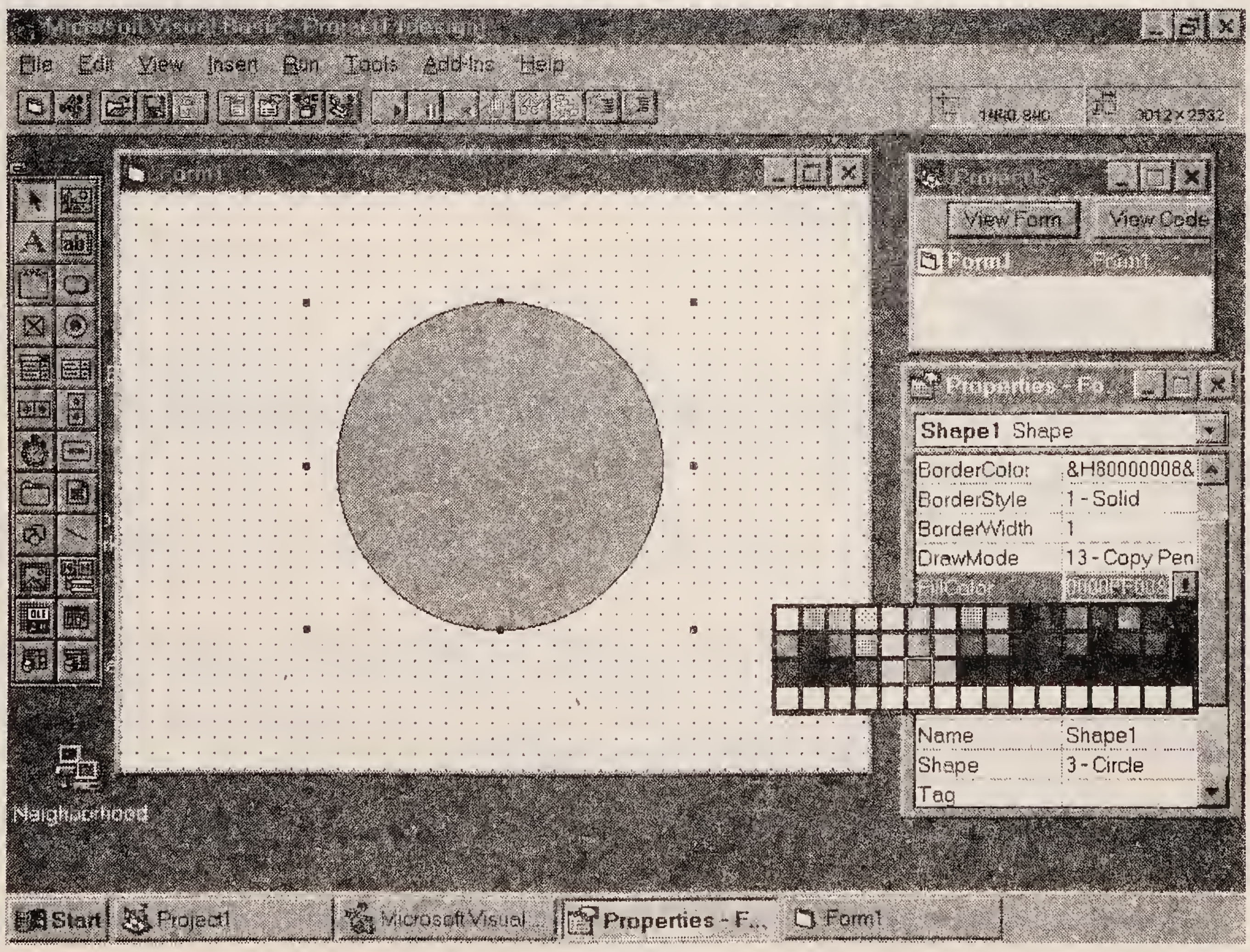
Properties - Form1

Shape8 Shape

BackColor	&H80000005&
BackStyle	0 - Transparent
BorderColor	&H80000008&
BorderStyle	1 - Solid
BorderWidth	1
DrawMode	13 - Copy Pen
FillColor	&H0000000
FillStyle	7 - Diagonal C
Height	972
Index	
Left	4560
Name	Shape8

图9.22　不同 FillStyle 属性值的 Shape 控件

运用 FillColor 属性可以为 Shape 控件着色。FillColor 属性适用于 Form、图片框、Printer 对象以及 Shape 控件。确定填充在 Shape 控件或用 Circle 和 Line 图形方法建立的圆形和矩形的颜色。在缺省时，FillColor 设置成0（黑色）。在 Shape 控件下可以用 Properties 窗口中的 FillColor 属性，在弹出的对话框中选择颜色。如图9.23所示。在代码中可用 RGB 或 QBColor 函数设置颜色。


Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
1480, 840
3012 x 2532
Form1
Project1
View Form
View Code
Form1
Neighborhood
Start
Project1
Microsoft Visual ...
Properties - F...
Form1

Properties - Form1

Shape1 Shape

BorderColor	&H80000008&
BorderStyle	1 - Solid
BorderWidth	1
DrawMode	13 - Copy Pen
FillColor	[illegible]
Name	Shape1
Shape	3 - Circle
Tag	

图9.23　从 FillColor 对话框中选择颜色填充控件

在这里需注意，当 FillStyle 属性值设置成1时，FillColor 的设置将被忽略。因为 FillStyle 的缺省值为1，即透明的。所以要想用 FillColor 属性为控件填充颜色，就首先得改变控件的 FillStyle 属性，使之不为1。

为 Shape 控件填充颜色除了可以使用 FillStyle 和 FillColor 属性以外，还可以使用 BackColor 和 BackStyle 属性。

BackColor 属性决定了 Shape 控件的背景色，可以通过 Properties 窗口中的 BackColor 属性中的设置框直接为其进行赋值，也可以通过颜色对话框直接选取某种颜色，并且这种颜色所对应的十六进制数值立即在设置框中显示出来。用户也可以在程序运行时通过在代码中使用 RGB()和 QBColor()函数来确定颜色。

在这里，如果 Shape 的 BackStyle 为(0)即透明色，则 BackColor 属性将被忽略，这同 Shape 的 FillColor 与 FillStyle 属性之间的关系相似。不过，Shape 的 BackStyle 属性只有两种属性。即 BackStyle＝0，Shape 控件为透明的，在控件后的背景色和任何图形都是可见的。BackStyle＝1，Shape 控件为不透明的(缺省值)，用控件的背景色填充控件并隐藏它后面的所有颜色和图形。

用户在使用这几种属性时也许发现了，当我们将 Shape 控件的 FillStyle 属性设置为0即实心，并将 FillColor 属性设置成某一种颜色时，无论怎样为 BackStyle 及 BackColor 属性赋值，其颜色特征都不会在 Shape 控件上显现出来，只有将 FillStyle 设置为1即透明时，BackColor 所设置的颜色才显现出来。这是因为 BackColor 只是为 Shape 控件设置了背景色，而 FillColor 设置的颜色将覆盖 BackColor 设置的背景色，只有当 FillStyle(透明)时才显现出来。

用户还可以使用 BorderColor 属性为 Shape 控件边界设置颜色。用 BorderStyle 属性为 Shape 控件的边界设置长划线方式，可以为实心线、透明线、长划线、点、点一划线、长划一点一点一长划线以及内部实线。可以用 BorderWidth 属性决定 Shape 控制边界的宽度。注意，BorderStyle 属性的设置效果依赖于 BorderWidth 特性的设置。若 BorderWidth 不为1，且 BorderStyle 不为0或6，则 BorderStyle 被设置为1。关于这三种属性，我们在介绍 Line(直线)控件时已经介绍过，这里就不再介绍了。

9.2.4 用 Circle 方法绘图

除了可以用 Shape 控件在窗体内形成圆、矩形等图形外，用户还可以使用 Circle 方法在对象中画圆、椭圆或弧，格式为：

[对象名.]Circle[Step](x,y),radius[,[Color][,[start][,end][,[aspect]]]]

各部分的描述如下表所示：

部分	描述
对象名	画圆所在的对象(Form、图片框和 Printer 对象)
Step	说明圆、椭圆或弧的中心，是相对于由 CurrentX 和 CurrentY 属性所指定的当前坐标所决定的
(x,y)	单精度数，指出圆、椭圆或弧的中心坐标，对象的刻度属性(ScaleMode,ScaleLeft 等)决定了所有的度量单位

（续表）

部分	描述
对象名	画圆所在的对象(Form、图片框和 Printer 对象)
radius	单精度数，指出圆、椭圆或弧的半径。对象的刻度属性(ScaleMode，ScaleLeft 等)决定了变量单位
Color	长整数，指出圆轮廓线的 RGB 颜色。省略时使用 ForeColor 属性值。可使用 RGB 函数或 QBColor 函数指定颜色
Start end	单精度数，当画部分圆或部分椭圆时，Start 和 End 指定弧的起始位置和终止位置(弧度形式)。范围是从-2π到2π弧度。Start 缺省值为0弧度，End 缺省值为2π弧度
aspect	单精度数，指出圆的纵横比。缺省值为1.0，即可在任何屏幕上画完美的圆

画好一个圆后，用户可以使用圆所在对象的 FillColor 和 FillStyle 属性。只有封闭的图形才能被填充。图形包括圆、椭圆或饼图(带有两条半径的弧)。画圆或椭圆的一部分时，如果 Start 为负值，Circle 至 Start 画一条半径，角度为正值，如果 end 为负值，Circle 至 end 处画一半径，角度当作正值。此方法绘图方向为逆时针。

用来画圆、椭圆或弧的线宽依赖于 DrawWidth 属性。在背景上画圆的方法依赖于 DrawMode 和 DrawStyle 属性的设置。

当画饼图形时，如果要在0度处画一条从左至右的半径，可指定 Start 为一很小的负值，而不是0。

用户可以省略语法中的参数，但必须在下一参数前输入一逗号。若省略一系列参数，则在指定最后一个参数时不用使用逗号。

下面，我们就举几个例子，示意一下怎样画圆、弧与饼图和椭圆。

(1) 本例在窗体中心画一同心圆。首先选择 File 菜单中的 New Project 命令。在打开的 Code 窗口中输入下列代码：

```
PrivateSub Form _ Click()
      Dim X，Y，I
      ScaleMode=3
      X=ScaleWidth/2
      Y=ScaleHeight/2
      If X>Y Then Limit=Y Else Limit=X
      For I=1 To Limit Step 2
        Circle (x,y) ,I, RGB(Rnd * 255，Rnd * 255，Rnd * 255)
    Next I
End Sub
```

单击工具条中的 Start 命令，在随后出现的 Form 窗口中单击鼠标，即在 Form 窗口中出现一组同心圆，如图9.24所示。

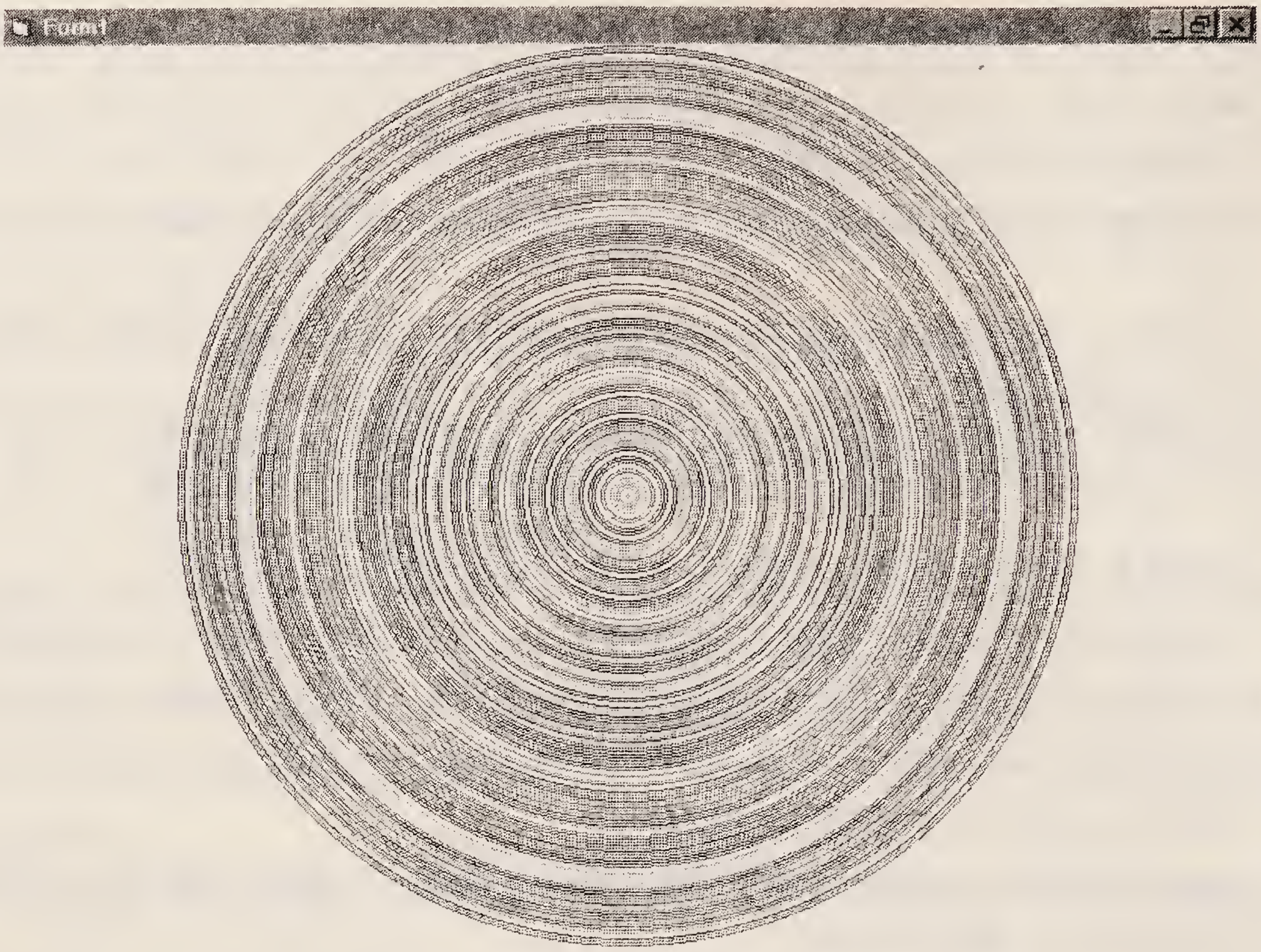


图9.24 Form 窗口中的一组同心圆

（2）本例在 Form 窗口中画一弧及一饼图。要想画一饼图，则在设置 Start 和 End 弧度值时，必须都用负值。在 Code 窗口中输入下列代码：

```
Private Sub Form _ Click()
     Dim XL, XR, Y, I,Pie
     ScaleMode=3:Pie =4 * Atn(1)
     XL=ScaleWidth/4 :XR=3 * ScaleWidth/4
     Y=ScaleHeight/2
     If XL>Y Then I=Y Else I=XL
     Circle (XL,Y),I,, -Pie/4, -3 * Pie/4
       If XL>Y Then I=Y Else I=XL
       Circle (XR,Y),I,, +Pie/4,+3 * Pie/4
End Sub
```

运行此程序，并在出现的 Form 窗口中单击鼠标，即出现一弧及饼图，如图9.25所示。

（3）本例在 Form 窗口中画两个宽高比不同的椭圆。只要在 Code 窗口中对2中的代码稍微修改即可：

```
Private Sub Form _ Click()
     Dim XL,XR, Y, I
     ScaleMode=3
     X1=ScaleWidth/4:X2=3 * ScaleWidth/4
     Y=ScaleHeight/2
     If XL>Y Then I=Y Else I=XL
     Circle (X1,Y), I, , , , .025
     If XL>Y Then I=Y Else I=XL
     Circle (X2, Y), I, , , ,4
End Sub
```

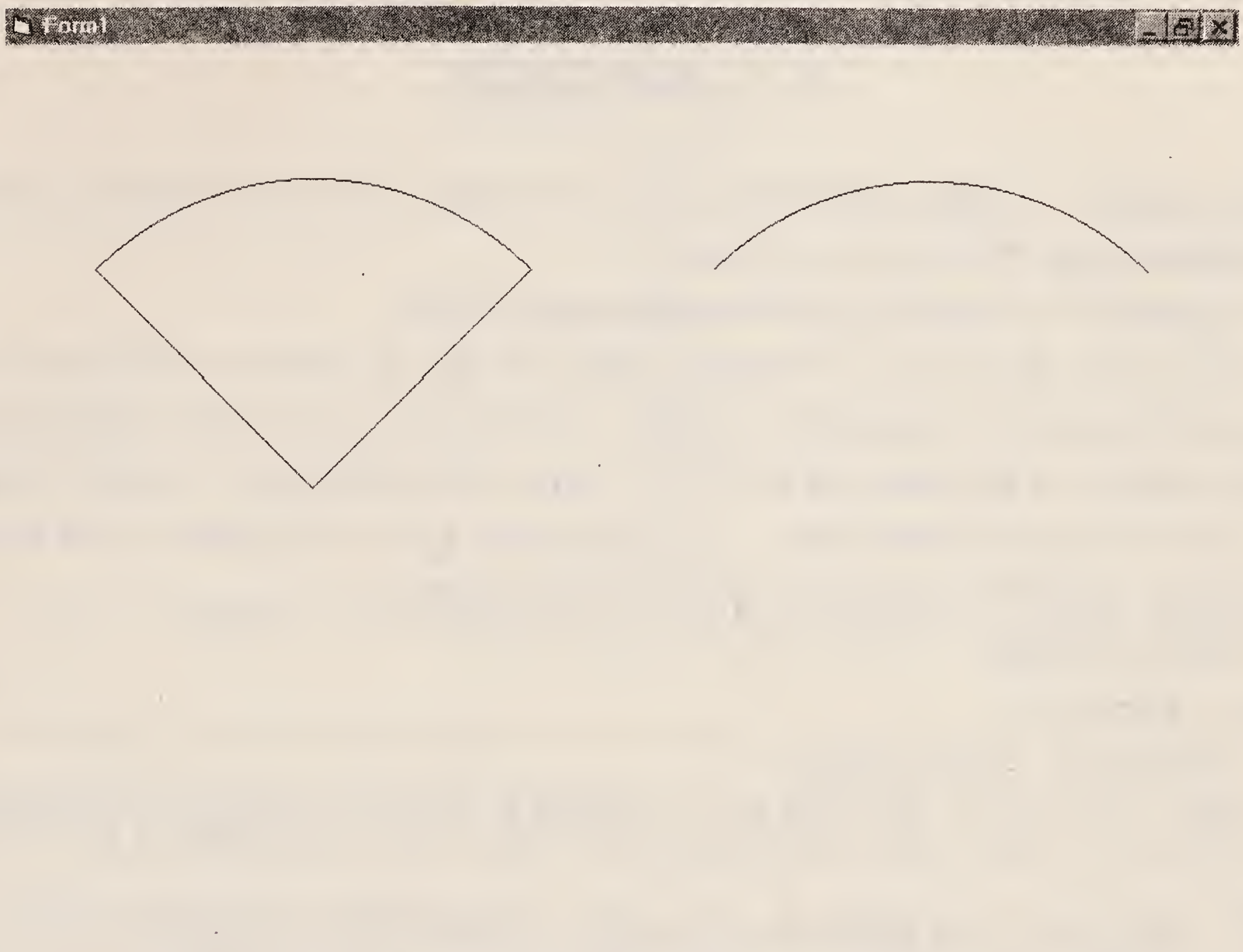


图9. 25　Form 窗口中的弧及饼图

在这段代码中我们加入了 aspect 值，不同的 aspect 值所画成的椭圆形状不同。在单击工具栏中的 Start 命令后，在出现的 Form 窗口中单击鼠标左钮，即出现两个不同形状的椭圆，如图9. 26所示。

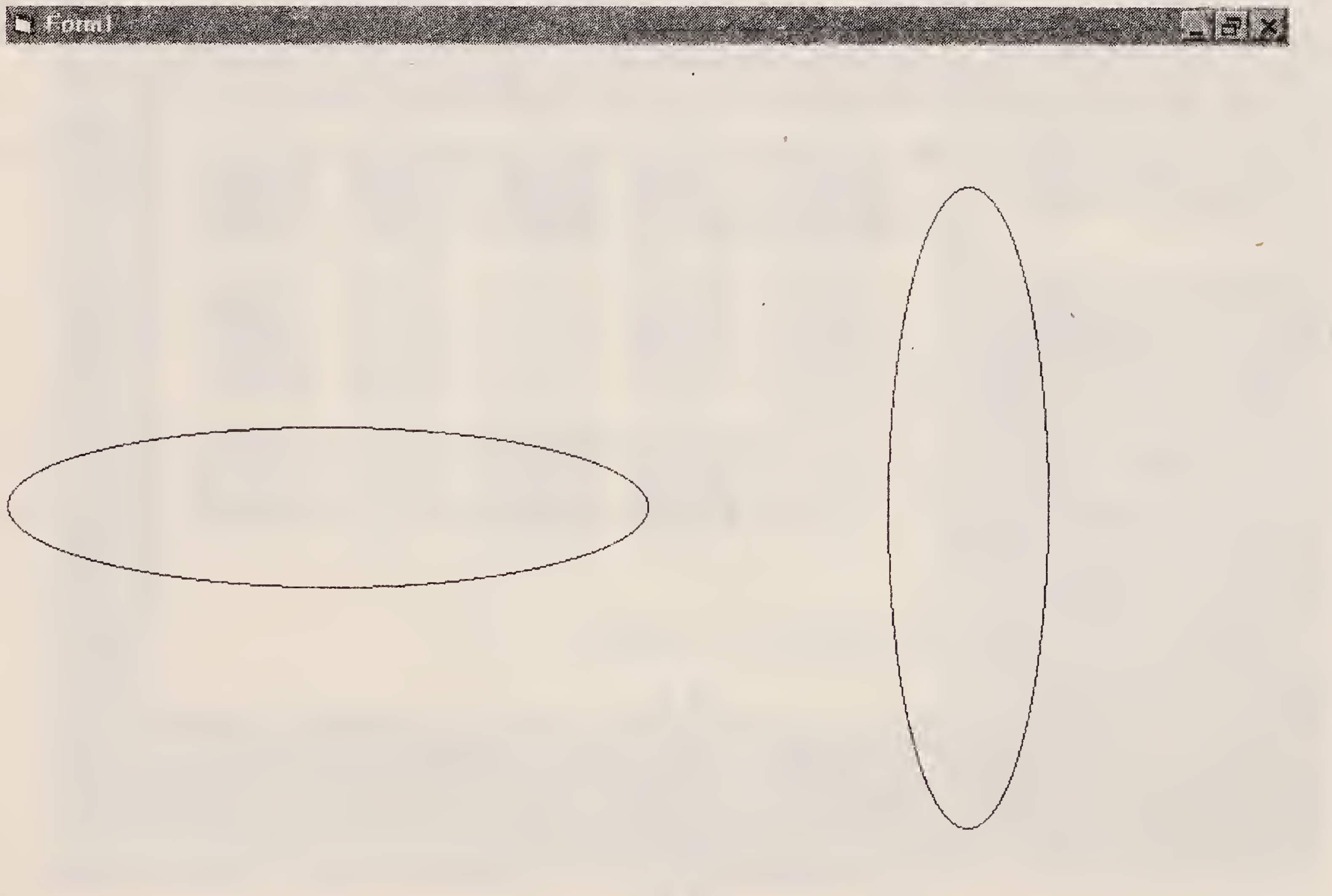


图9. 26　不同 aspect 值对应的椭圆形状不同

9.3 颜色的使用

用户在绘图时，一般都要为图形着色。如果不指定颜色，Visual Basic 将对所有的图形方法使用对象的前景色，指定色彩有三种方法。

（1）直接使用十六进制代码或选择对象颜色属性对话框。

关于这种方法，用户应该已经很熟悉了。我们在前面介绍 BackColor 或 BorderColor 等属性时已介绍过这种方法，就是当用户选定某一对象后，在 Properties 窗口中选择它的某一颜色属性，即在这一属性右侧的设置框中出现一按钮，同时设置框中的十六进制代码将被涂以高亮。用户可以直接在设置框中输入一十六进制代码，也可以单击按钮，在出现的颜色对话框中选择其一颜色即可。这种方法一般是为一固定对象着色。

（2）使用 RGB 函数

RGB()函数格式为：

RGB(red%,green%,blue%)

在这里 red%,green%,blue%是0到255之间的整数，代表每一种组成混合色的颜色的强度。

RGB()函数返回由三种参数经混合后组成的一个长整数。这个长整数在 Visual Basic 解释中表示一个特定的混合颜色值。哪个参数值越大，则在混合色中该颜色越亮。

关于红、绿、蓝这三种颜色，用户应该很熟悉了，我们计算机显示屏以及电视机等都是使用这三种颜色作为三元色。这三种颜色各使用不同的强度可以混合成许多种颜色。例如，用红色和绿色，混合可以得到黄色，用 RGB()函数表示，可以为 RGB(255,255,0)，在这里红色和绿色达到最亮值，没有蓝色，经混合后可以组成最亮的黄色。用红、绿和蓝三种颜色可以组成白色，RGB 函数为 RGB(255,255,255)。

用 RGB()函数确实可以组成许多种颜色，但并不能组成所有的颜色。所以用户在使用 RGB()函数时对于一些并不太常用的颜色，如叶绿色，暖灰色等就不必太强求。

（3）QBColor()函数

QBColor()函数格式为：

QBColor(ColorCode%)

在这里 ColorCode%为0～15之间的整数，不同的整数值代表不同的颜色。QBColor()函数为 Visual Basic 解释器返回一个代表混合颜色的长整数。下面我们就列出不同 ColorCode%值所代表的颜色。

QB 颜色值	QB 颜色
0	黑色
1	蓝色
2	绿色
3	深青色
4	红色
5	品红色
6	深黄色

（续表）

QB 颜色值	QB 颜色
7	纸白色
8	灰色
9	亮蓝色
10	亮绿色
11	亮青色
12	亮红色
13	亮品红色
14	亮黄色
15	亮白色

下面一个示例程序，在 Form 窗口中画出16个图片框控件，然后将其设置成16种不同的颜色。

首先选择工具框中的图片框控件，在 Form 窗口中画出16个图片框，然后在每个图片框控件上方画一标签控件，将每个标签控件的 Caption 属性设置为 QBColor(i)，其中 i 为对应某个图片框控件的颜色值。如对应第一个图片框的标签，将其 Caption 属性改为 QBColor(0)，即表示该图片框中的颜色为黑色。单击 Project 窗口中的 View Code 按钮，将每个图片框的 BackColor 属性用 QBColor()函数赋值。例如 Picture1. BackColor＝QBColor(0)，Picture 2. BackColor＝QBColor(1)等等。输完代码后，单击工具栏中的 Start 命令，单击 Form 窗口，则每个图片框的颜色将变为其 QBColor()函数所对应的颜色，如图9. 27所示。

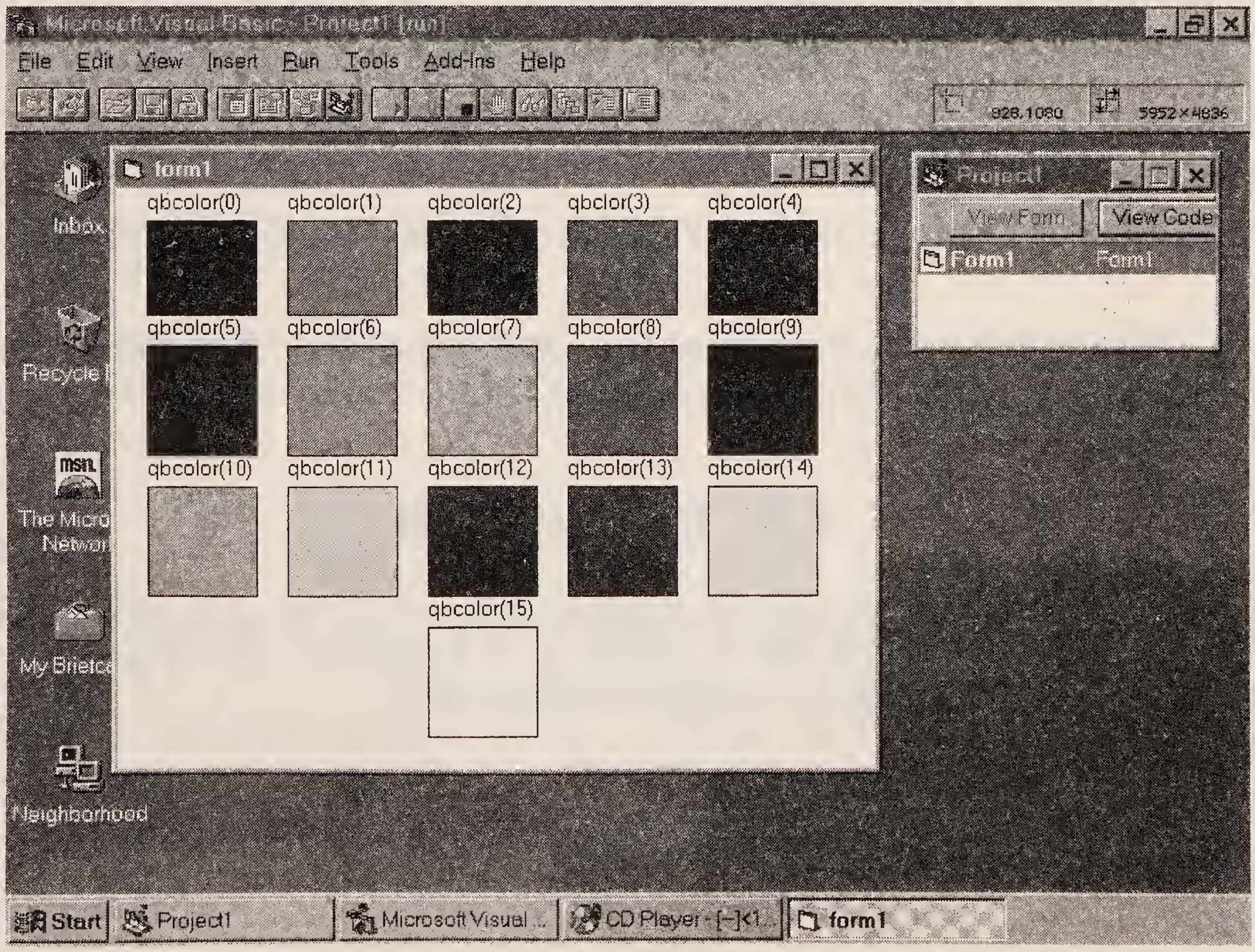

Microsoft Visual Basic - Project1 [run]
File Edit View Insert Run Tools Add-Ins Help
828,1080
5952×4836
form1
qbcolor(0)
qbcolor(1)
qbcolor(2)
qbclor(3)
qbcolor(4)
qbcolor(5)
qbcolor(6)
qbcolor(7)
qbcolor(8)
qbcolor(9)
qbcolor(10)
qbcolor(11)
qbcolor(12)
qbcolor(13)
qbcolor(14)
qbcolor(15)
Project1
View Form
View Code
Form1
Form1
Inbox
Recycle
The Micro Networ
My Briefc
Neighborhood
Start
Project1
Microsoft Visual ...
CD Player - [--]<1...
form1

图9. 27　不同 QBColor 函数值对应不同的颜色

9.4　定制标度

在前面，当我们要把图形的中心放在 Form 窗口的中心时，使用的是 CX＝ScaleWidth/2 以及 CY＝ScaleHeight/2语句。这是因为 Visual Basic 通常把Form 窗口的左上角的坐标定义为(0,0)。这就像一个倒置的笛卡尔坐标，即 x 轴水平指向 Form 右侧，y 轴垂直指向 Form 下边，坐标原点在 Form 窗口左上角。这样的坐标对于大多数用户来说并不太适应。用户可以用 Scale 方法来重新定制标度，Scale 方法可以为 Form 或图形设置新的坐标系统，且新的坐标系统可以适用于任何图形方法。

例如：Scale(－200,100)－(200,－100)

它设置了一个新的坐标系统，其左上角坐标为(－200,100)，右下角坐标为(200,－100)，按顺时针方向为 Form 或图形四角设置的坐标分别为：

(－200,100)

(200,100)

(200,－100)

(－200,－100)

这时，(0,0)就代表 Form 或图形的中心点，且 x 轴水平指向右侧，y 轴垂直向上。这种转换的产生是由于 Visual Basic 的 Scale 方法将标准标度改为定制标度了。

一般地，Scale 方法的格式如下：

```
Scale(LeftX,TopY)－(RightX, BottomY)
```

在这里 LeftX 代表最小 X 坐标值(左侧)，TopY 代表最大 Y 值(顶端)，RightX 为最大 X 值(右侧)，BottomY 代表最小 Y 值(底端)，它们都为单精度实数。

例如：Scale(－1E38, 1E38)－(1E38,－1E38)给出了最大可能的标度，可以定位很小的细节。而 Scale(－1,1)－(1,－1)则给出了相对较小的标度，所以邻近像素的坐标的变化很细微。

如果不带坐标使用 Scale 方法，Visual Basic 将复位坐标系统。复位后标度为缺省方式(0,0)代表左上角，单位为 twip(1twip＝1/1440英寸)。

9.5　PSet 方法绘图

现在我们已经知道了怎样设置颜色并根据用户需要设置标尺。那么怎样控制像素呢?我们可以使用 PSet 方法控制像素并可以使其具有特定的颜色。

PSet 方法的格式如下：

```
[object.]PSet[Step](x,y)[,color]
```

方框内的设置可以省略，其中各部分的作用如下表所示：

部分	描述
object	Form，图片框，Printer 对象，在其上画点
Step	关键字，规定了坐标是同由 CurrentX 和 CurrentY 属性所决定的图像位置相关的
x，y	单精度值，指示设置点的横坐标和纵坐标
color	RGB 决定点的颜色，若缺省，则由前景颜色确定，可由 RGB 和 QBColor 函数来指定颜色

用 PSet 方法为对象上某一点设置颜色时，点的位置由当前选定的标尺(scale)来决定。如果用户选定的某一点位于对象之外时，将 AutoRedraw 属性设置为 Ture，Visual Basic 将记录这一信息，但并不绘制这一点。如果绘制的某一点位于窗体之外时，将会导出溢出错误。所以用户在选定某点坐标时，首先要注意一下这时的 ScaleMode 值，避免出现溢出错误。

刚才说过，如果将 Form 的 AutoRedraw 属性设置为 True 时，Visual Basic 将记录选定的 Form 之外但仍在屏幕中的选定点的信息，且当 Form 窗口最大化时立即显现出来。

下面我们就举个例子，在一有限 Form 窗口内画一组平行线，一组垂直线，并且这些线将遍布整个屏幕。当然在有限 Form 窗口中由于 Form 窗口的限制，只能显示一部分，当 Form 窗口最大化时，才全部显示出来。

步骤是：单击 File 菜单中的 New Project 命令，然后在 Code 窗口中输入下列代码：

```
Private Sub Form _ Click()
    ScaleMode =3
    Dim CX As Integer, CY As Integer, I As Integer
    AutoRedraw = True
    For CX=0 TO 600 Step 50
        For CY=0 TO 500
            PSet(CX, CY),QBcolor (CX/30 Mod 15)
        Next CY
      Next CX
    For CY =0 To 500 Step 50
  For CX=0 To 600
      PSet(CX,CY),QBColor(CY/60 Mod 15)
    Next CX
  Next CY
End Sub
```

由于程序循环太长，也许要等一段时间结果才能显示在屏幕上，读者不要着急，也不要认为程序有错，笔者在 Visual Basic 4.0上已运行过这一程序，结果是正确的。

输完程序后，运行程序，并单击 Form 窗口，即出现一组平行线与垂直线组成的网格，如图9.28所示。

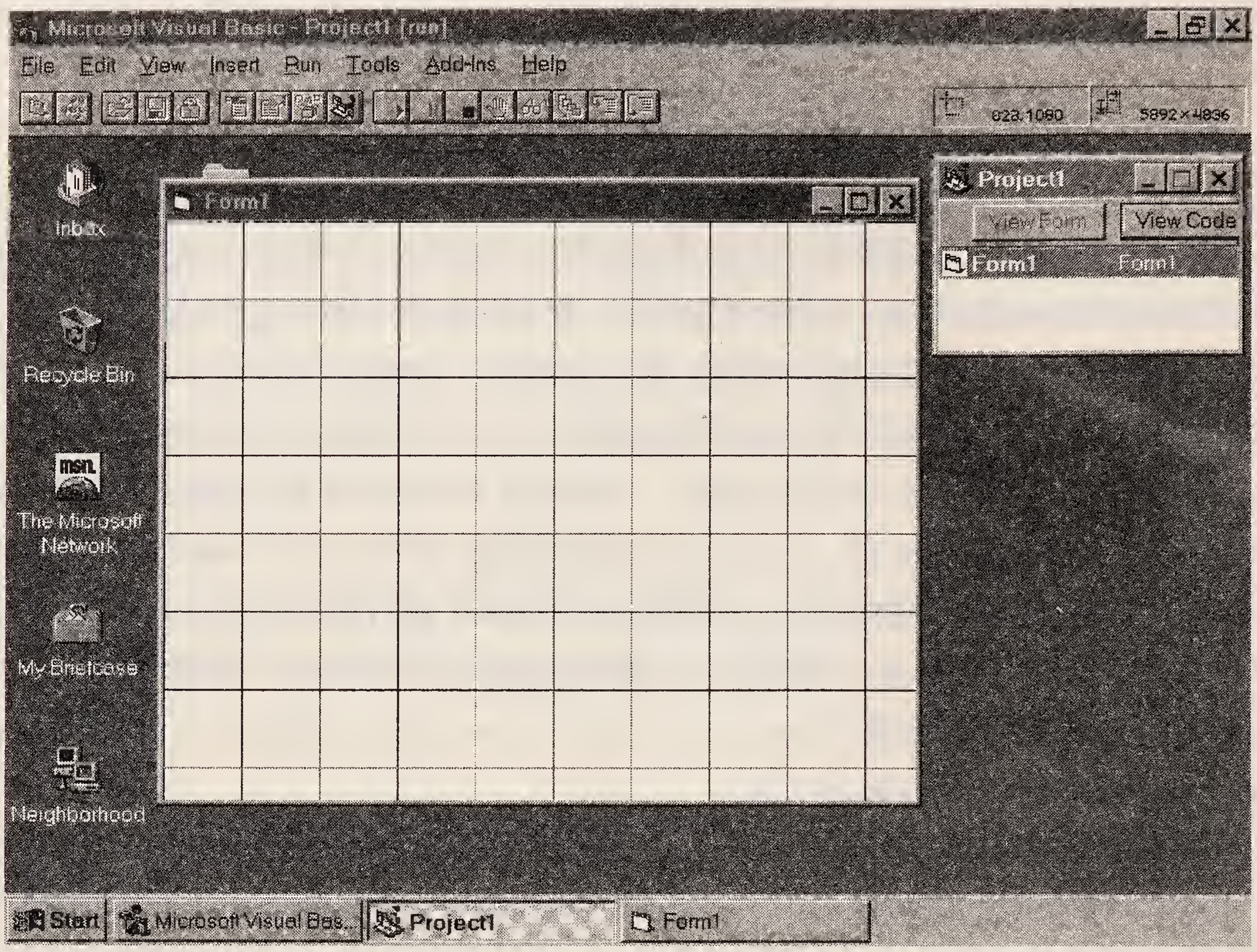

Microsoft Visual Basic - Project1 [run]
File Edit View Insert Run Tools Add-Ins Help
823,1080
5892×4836
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Neighborhood
Form1
Project1
View Form
View Code
Form1
Form1
Start
Microsoft Visual Bas...
Project1
Form1

图9.28 PSet 画出的网格图形(其中部分被隐含)

然后,单击 Form 窗口的放大图标(在 Form 窗口右上角三个图标的中间那个),则在最大化的 Form 窗口中显现出被放大前 Form 窗口截断的图形。如图9.29所示。

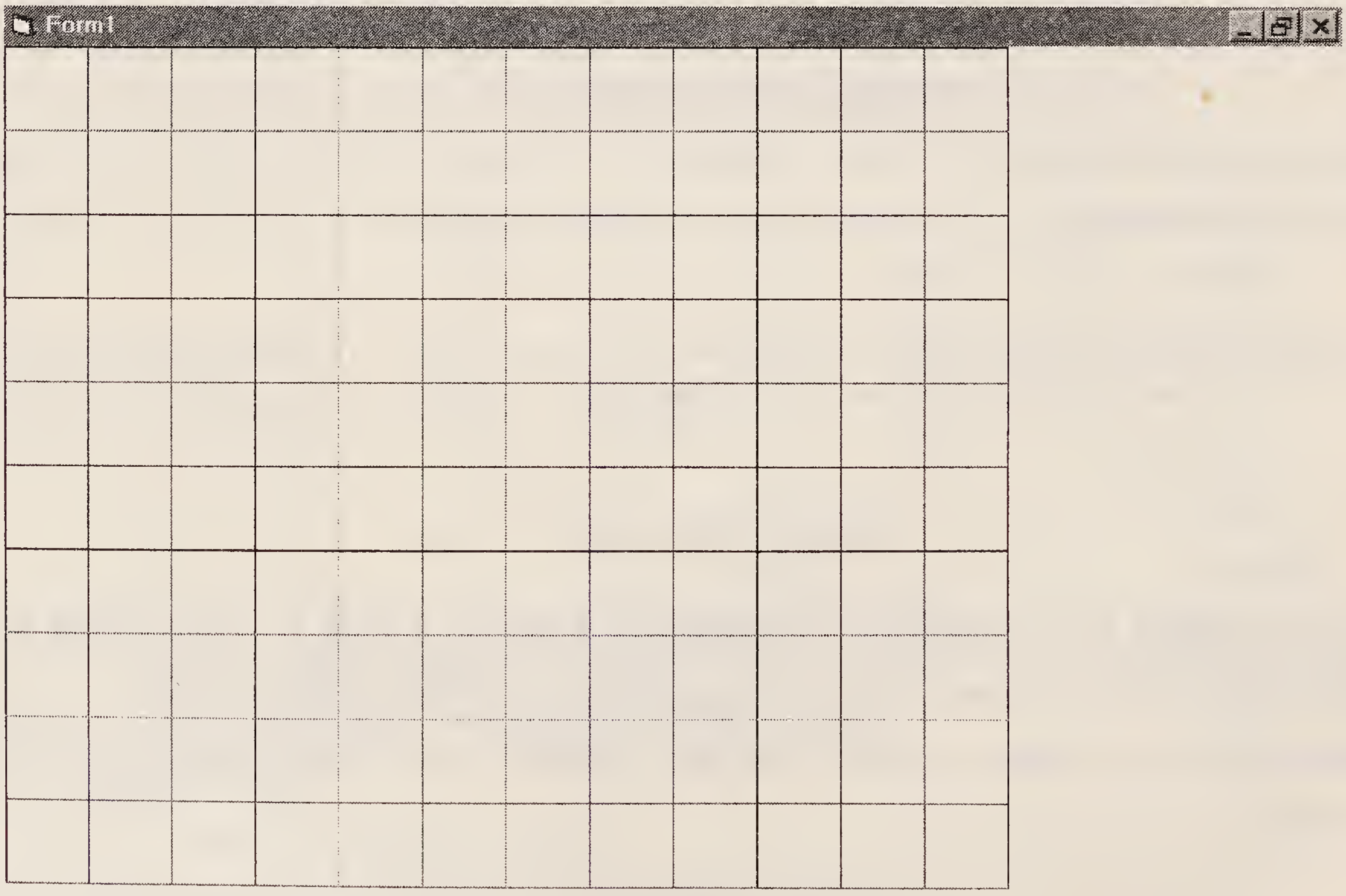


图9.29 图9.28中未被显现部分的显现

用PSet方法除了可以给其指定点着色外，还可以将某指定点擦去。其实就是将指定点以背景色重画。

例如，我们在上述程序最后加入另一循环，这一循环将擦掉水平线上的某些点。

```
For CY=0 To 500 Step 50
  For I=0 To 580 Step 20
    For CX=I+10 To I+20
      Pset(CX,CY), BackColor
    Next CX
  Next I
Next CY
```

这时运行程序时，我们会看到Form窗口中的水平线已不再是实线，而变为虚线。这是由于某些点被以背景色重画，相当于被擦去的结果。如图9.30所示。

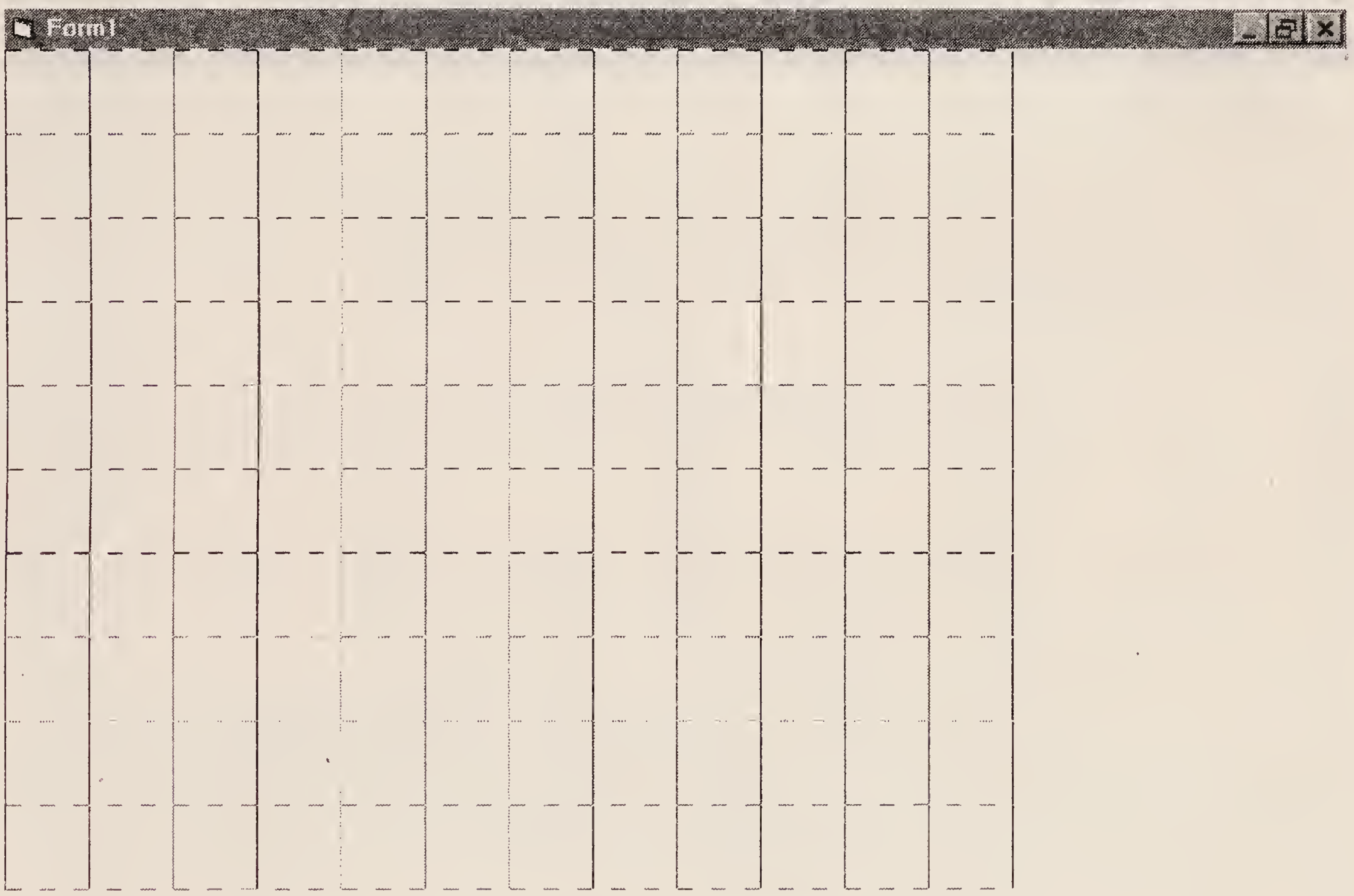


图9.30　Form窗口中水平线的某些点被擦去

由于Visual Basic对不同的坐标的处理方式不同，也可能在某些监视器上不能正常执行，并且也许还会出现未擦或误擦的现象，可以使用较大的Step(步长)解决这个问题。

9.6　绘制曲线

用户在绘图时有可能会绘制一些曲线，如正弦线或一些优美的图形。下面我们就举两个绘制曲线的例子。

(1) 在Form窗口内画一条正弦线。

首先建立一个新的Project文件，将Form窗口的Caption属性定义为Sin Graph，然后打开Code窗口，输入下列代码：

```
Private Sub Form _ Click()
    Dim I As Single ,TwoPie As Single
    TwoPie = 8 * Atn (1)
    Scale (—TwoPie,1)—(TwoPie, —1)
    For I=—TwoPie To TwoPie Step .01
        PSet(I, Sin(I))
    Next I
End Sub
```

这段代码，首先对 TwoPie 进行赋值，在这里 Atn(1)表示 $\mathrm{arctg}(1)=\frac{\pi}{4}, 8*\frac{\pi}{4}=2\pi$。然后定制了新的标度，表示 Form 窗口的左上角坐标为$(-2\pi,1)$右下角坐标为$(2\pi,-1)$，Form 窗口中心为坐标原点(0,0)，最后用 PSet 方法绘制正弦曲线。

运行程序时，单击 Form 窗口，即出现一正弦图形，如图9.31所示。

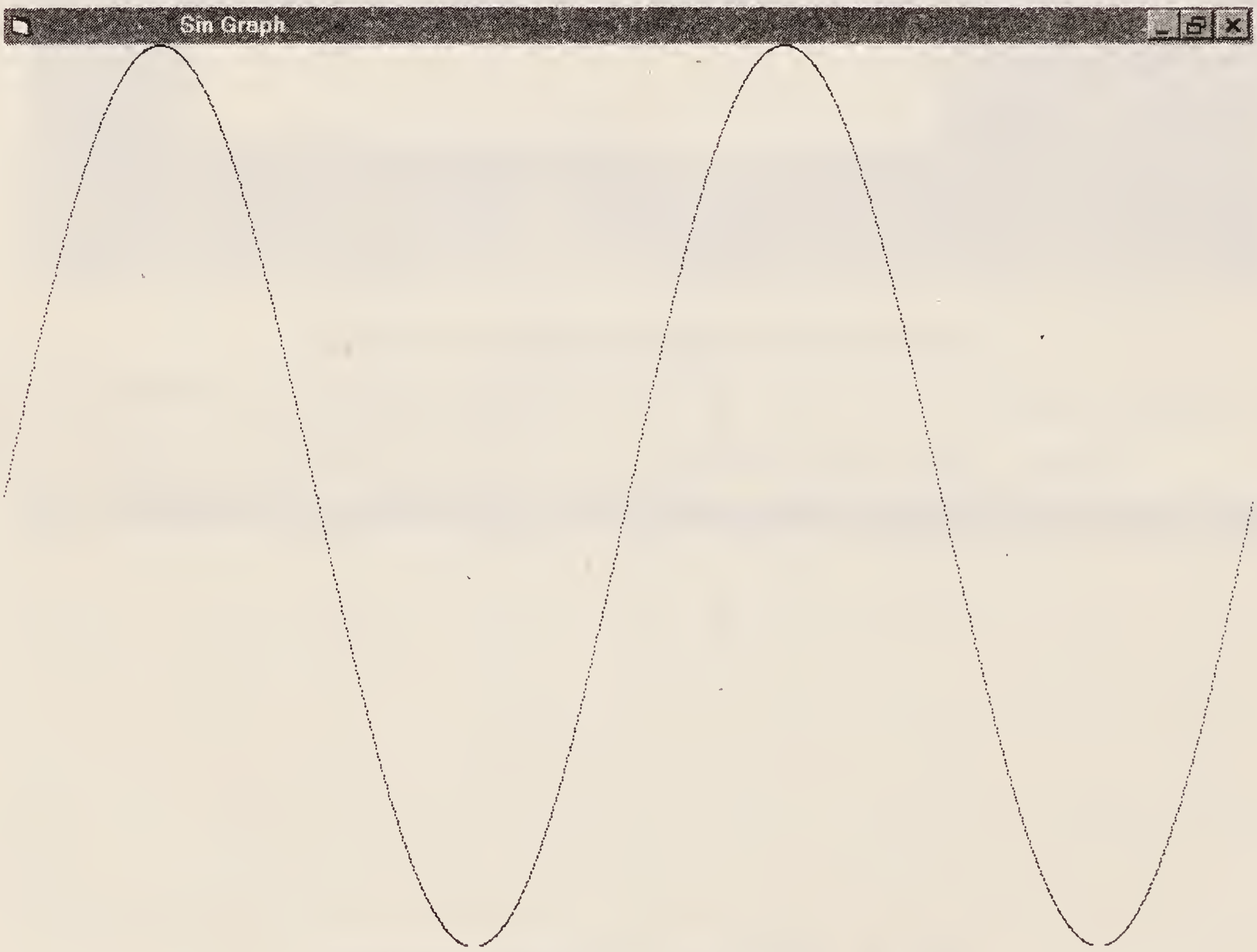


图9.31 正弦函数曲线

(2) 在 Form 中绘制由参数方程

$x=\sin 2t * \cos t \quad \theta \leqslant t \leqslant 2\pi$

$y=\sin 2t * \sin t$

所表示的函数曲线。

单击 File 菜单中的 New Project 命令。再单击 Project 窗口中的 View Code 命令，在 Code 窗口中输入下列代码：

```
Private Sub Form _ Click()
    Dim X As Single, Y As Single
    Dim TwoPie As Single , I As Single ,R As Single
    Scale (—2, 2)—(2,—2)
    Line (0,1)-(0,-1)
    TwoPie=8 * Atn(1)
    For I=0 To TwoPie Step.01
        R=Sin(2*I)
        X=R*Cos(I)
        Y=R*Sin(I)
    PSet(X,Y)
  Next I
End Sub
```

运行程序,单击 Form 窗口,将出现图9.32所示的图形。

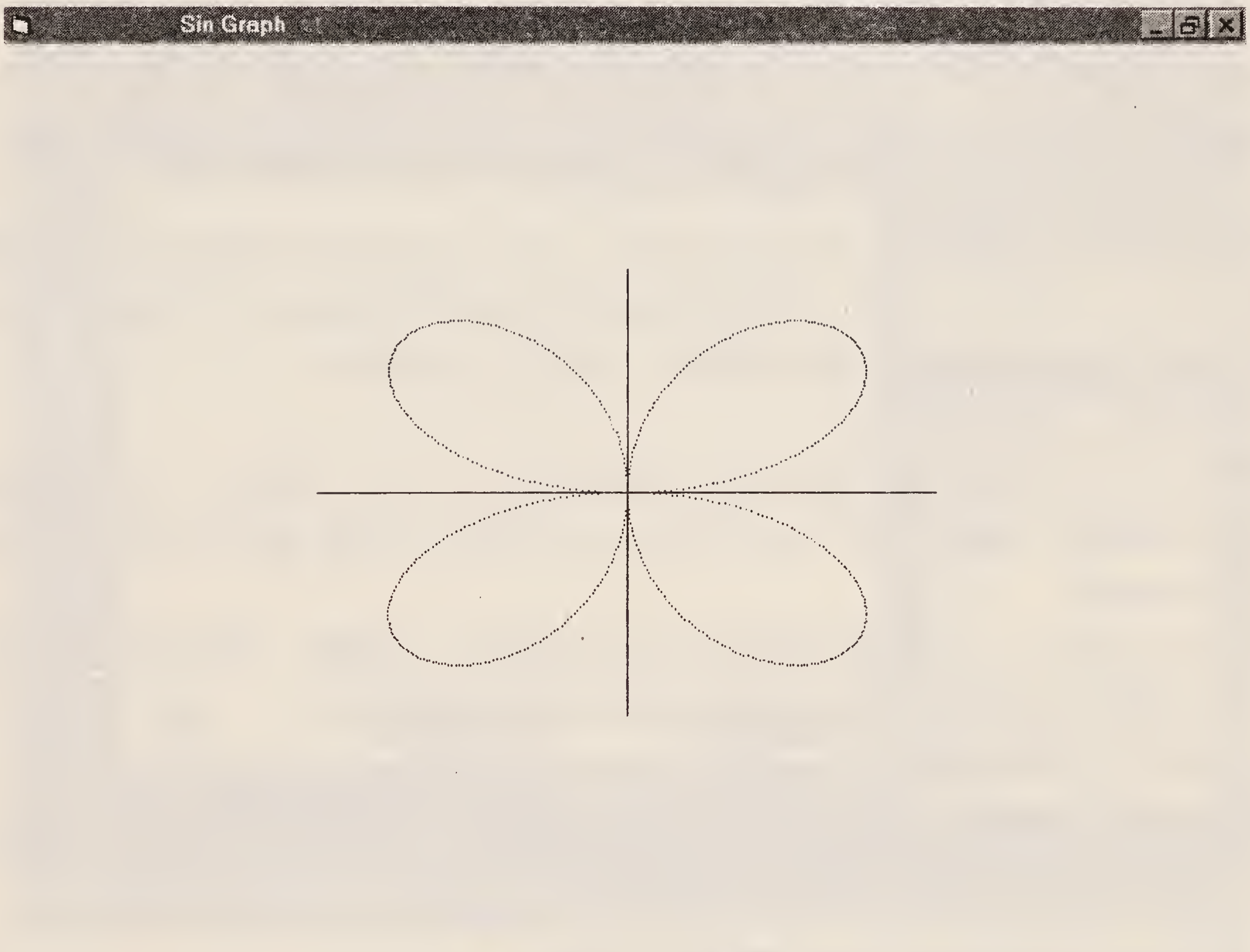


图9.32　一曲线图形

在绘制曲线时,一般应先知道这一曲线对应的方程式,在编制代码时,用户可以根据自己的习惯来定制新的标度。

9.7　简单动画设计

改变一个图标的形状或拖动一个对象经常令我们感到很有趣,但对一个程序员来说编制一个动画程序则更令人兴奋。动画其实就是在屏幕上快速地显示一系列相关的图像。我们在 Windows 中经常用到的拖放控件就是一个最简单的动画,它允许用户把一副图像从屏幕的一个地方移到另一个地方。而真正的动画是有计划地移动一个对象包括改变对象的形状及尺寸。在这一节我们将学习怎样在程序中建立一个简单的动画,怎样使用 Move 方法,改变图片框的 Top 和 Left 属性,以及用一个时钟(Timer)控件来控制动画的速度。

9.7.1 制定 Form 坐标系

动画中对象的移动是与 Form 中预定义的坐标系有关的。在 Visual Basic 中，每个 Form 都有它自己的坐标系。在缺省情况下，坐标原点(0,0)位于 Form 的左上角，度量单位是 twip | twip＝1/1440(英寸)，横坐标(即 X 轴)水平指向 Form 窗口右侧，纵坐标(Y 轴)垂直指向 Form 窗口下侧。用户可以通过定义坐标(x,y)来定义一个对象的放置位置。对于这种坐标系不太习惯的用户可以使用Scale 方法重新定义坐标系，这种方法在9.4节中我们已经讲过了，这里就不再重复。

9.7.2 在预定义坐标系中移动对象

Visual Basic 提供了一种 Move 方法，使用户可以在坐标系中移动某一对象。

Move 方法的格式为：

```
Object. Move Left, Top[Width, Length]
```

Object 是在 Form 窗口中将被移动的对象。Left 和 Top 则是在屏幕坐标系中对象被移到的新的位置。Left 值表示 Form 窗口的最左边与对象之间的距离，Top 值则表示 Form 窗口的最上边与对象之间的距离，在缺省情况下，其值为 twip。例如：

```
Picture1. Move 1000,1000
```

表示将 Picture1移到屏幕中新的位置，坐标为(1000,1000)。对象除了可以被移动以外，还可以改变其高度与宽度。通过 Width 和 Length 指定对象的新宽度与新长度，Width 和 Length 值可以省略。

用户还可以用 Move 方法来实现相对运动。即用加(＋)或减(－)操作改变某一对象的 Top 和 Left 属性，使其在原来的位置基础上移动到新的位置。例如：

```
Picture1, Move Picture1. Left－50, Picture 1. Top－50
```

即将 Picture1从当前的位置移动到一个新的位置。新位置距离 Form 窗口左侧将比原位置近50(twips)，距离 Form 窗口顶部将比原位置近50(twips)。

9.7.3 用 Move 方法和时钟(Timer)控件建立动画

在程序中添加动画程序，就是在时钟事件过程中添加一个或多个 Move 方法，使得对象在时钟的控制下在 Form 窗口中进行移动。在前面的窗体与工具中，我们已介绍过时钟控件的使用方法，时钟其实就是为程序运行提供准确的时间操作。用户可以通过改变时钟的 interval 属性来控制动画的运动快慢。另一方面，用户还得通过对象的 Top 和 Left 属性来确定对象的位置，以及判断对象是否已运动到窗体的边界来确定是否停止程序的运行。

9.7.4 动画设计

在这个动画程序里，我们将使一个地球图标在 Form 窗口内来回移动，其移动的速度由时钟控件 Timer 来进行控制。

其具体设计步骤如下：

(1) 单击 File 菜单中的 New Project 命令，把 Form1窗口的 Caption 属性设置为 my first animation。

（2）单击工具框中的图像框控件，在Form窗口的中部画一框体，并将其BorderStyle属性设置为0（即没有边框）。

（3）单击工具框中的Timer控件，在Form窗口的左下部绘一时钟控件，将其Interval属性设置为60。

（4）单击工具框中的Command Button控件，在Form窗口的中下部绘一命令按钮，其缺省Caption属性值为Command1，将其Caption属性设置成Begin，Name属性也设置成Begin。在Form窗口的右下角也绘制一命令按钮，将其Caption属性设置成End，Name属性也设置成End。

这样在Form窗口中的工作就完成了。这时的Form窗口如图9.33所示。

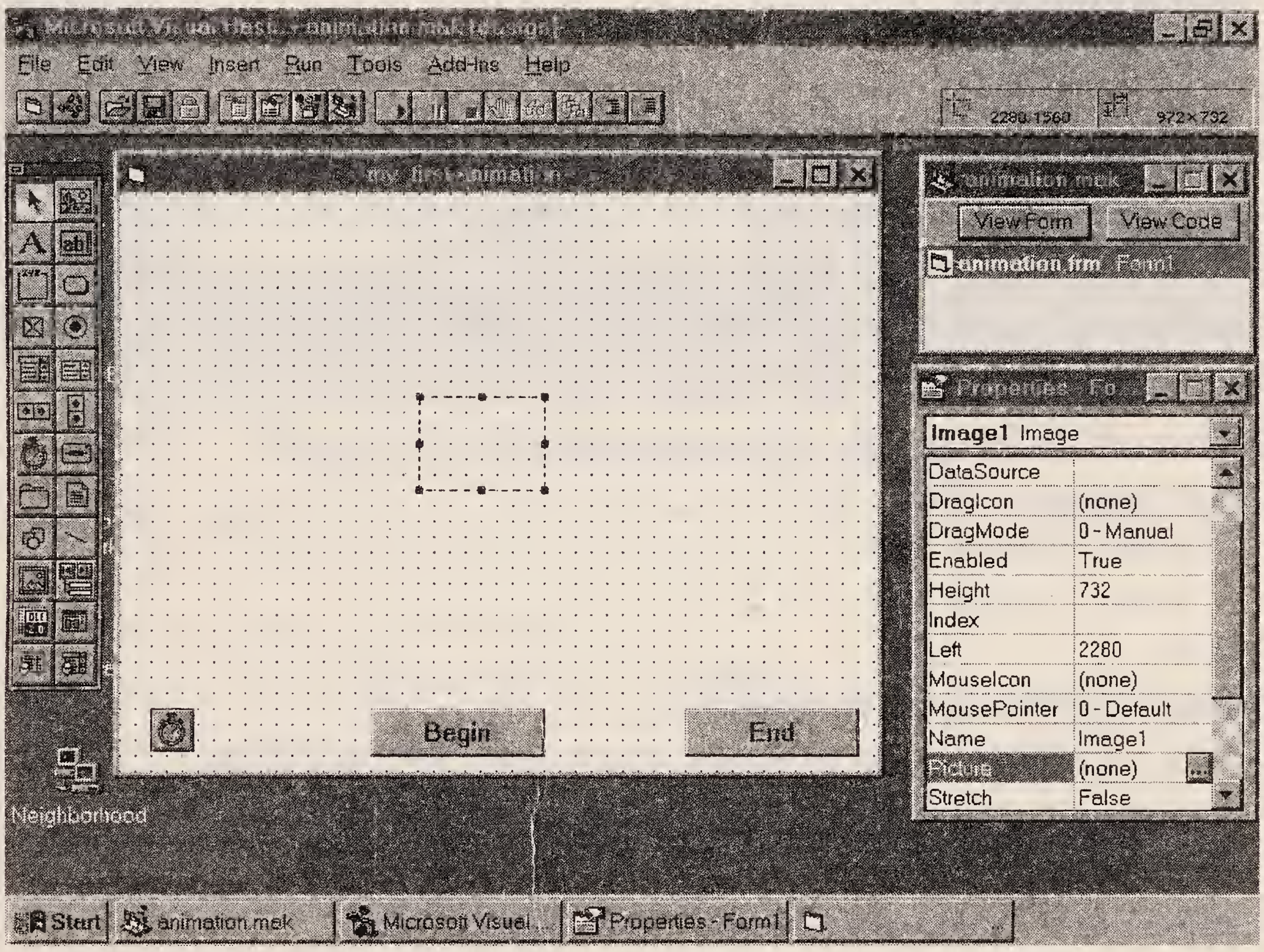

File Edit View Insert Run Tools Add-Ins Help
2280,1560
972×732
View Form
View Code
animation.frm Form1
Begin
End
Neighborhood
Start
animation.mak
Microsoft Visual...
Properties - Form1

Image1 Image

DataSource	
DragIcon	(none)
DragMode	0 - Manual
Enabled	True
Height	732
Index	
Left	2280
MouseIcon	(none)
MousePointer	0 - Default
Name	Image1
Picture	(none)
Stretch	False

图9.33　设置好的Form窗口

（5）下面所要做的工作就是为程序编写代码。通过Project窗口的View Code按钮打开代码窗口，在Object列表框中选择General，在窗口中输入下列代码来定义图像框每次移动距离的变量：

```
Dim CX As Integer, CY As Integer
```

然后选择Object列表框中的Form，Procedure列表框中的Load，输入下列代码。

```
Private Sub Form _ Load()
  CX=50:CY=75
  Image1.Visible=False
  Timer1.Enabled=False
  Image1.Picture=LoadPicture(″C:\vb\icons\elements\earth.ico″)
End Sub
```

这段代码的意思是首先给CX，CY变量赋值，然后使Image1图像框隐藏起来，将时钟控件设置为不可执行的，即Timer1不起作用，再为图像框装入earth（地球）图标。

再从 Object 列表框选择 Timer1，即出现 Timer1_ Timer()事件过程头与过程尾，输入下列代码：

```
Private Sub Timerl _ Timer()
      Image1. Move Image1. Left＋CX， Image1. Top＋CY
      If Image1. Left＋Image1. Width＞ScaleWidth＋ScaleLeft
            Then CX＝－50
      If Image1. Top＋Image1. Height＞ScaleHeight＋ScaleTop
            Then CY＝－75
      If Image1. Top ＜ ＝ 0 Then CY＝50
      If Image1. Left＜0 Then CX＝75
End Sub
```

这段代码的意思是在时钟控制下，每60ms(因为 Timer 的 interval 属性值为60)Image1 的 Left 属性将增加 CX 值，Top 属性增加 CY 值。当图像框运动到 Form 窗口的右侧。即 Image1的 Left 与 Width 值大于窗体 ScaleWidth 与 ScaleLeft 值时，将 CX 值变为－50，即让其向左运动。当图像框运动到 Form 窗口的底边界时，将 CY 值变为－75，使其向上运动。当图像框运动到 Form 窗口的顶部时，将 CY 值变为75，使其向下运动。当图像框运动到 Form 窗口的左侧时，则使 CX 值变为50，使其向右运动。

选择 Object 列表框中的 begin 与 Procedure 列表框中的 Click，输入下列代码：

```
Private Sub begin _ Click()
      Image1. Visible ＝ True
      Timer1. Enabled＝True
End Sub
```

使图像框显现出来，时钟控件开始动作。

选择 Object 列表框的 end 与 Procedure 列表框中的 Click，输入代码：

```
Private Sub end _ Click()
      End
End Sub
```

来结束程序的运行。

输完程序后，关闭 Code 窗口，并选择 File 菜单中的 Save Project 命令保存工程文件。

单击工具栏中的 Start 按钮运行程序，单击 Form 窗口中的 Begin 按钮，即出现一地球在 Form 窗口内像一皮球一样来回移动，且碰到 Form 窗口边界时发生碰撞，被弹了回来。如图9.34所示。读者可以上机试试这个程序的操作，非常有趣。

我们在介绍图像框控件时曾介绍过它有一种 Stretch 属性，可以使图像框中的图形放大或缩小来适应图像框。下面我们就利用这一属性使地球图标在运行过程中不断地放大和缩小。

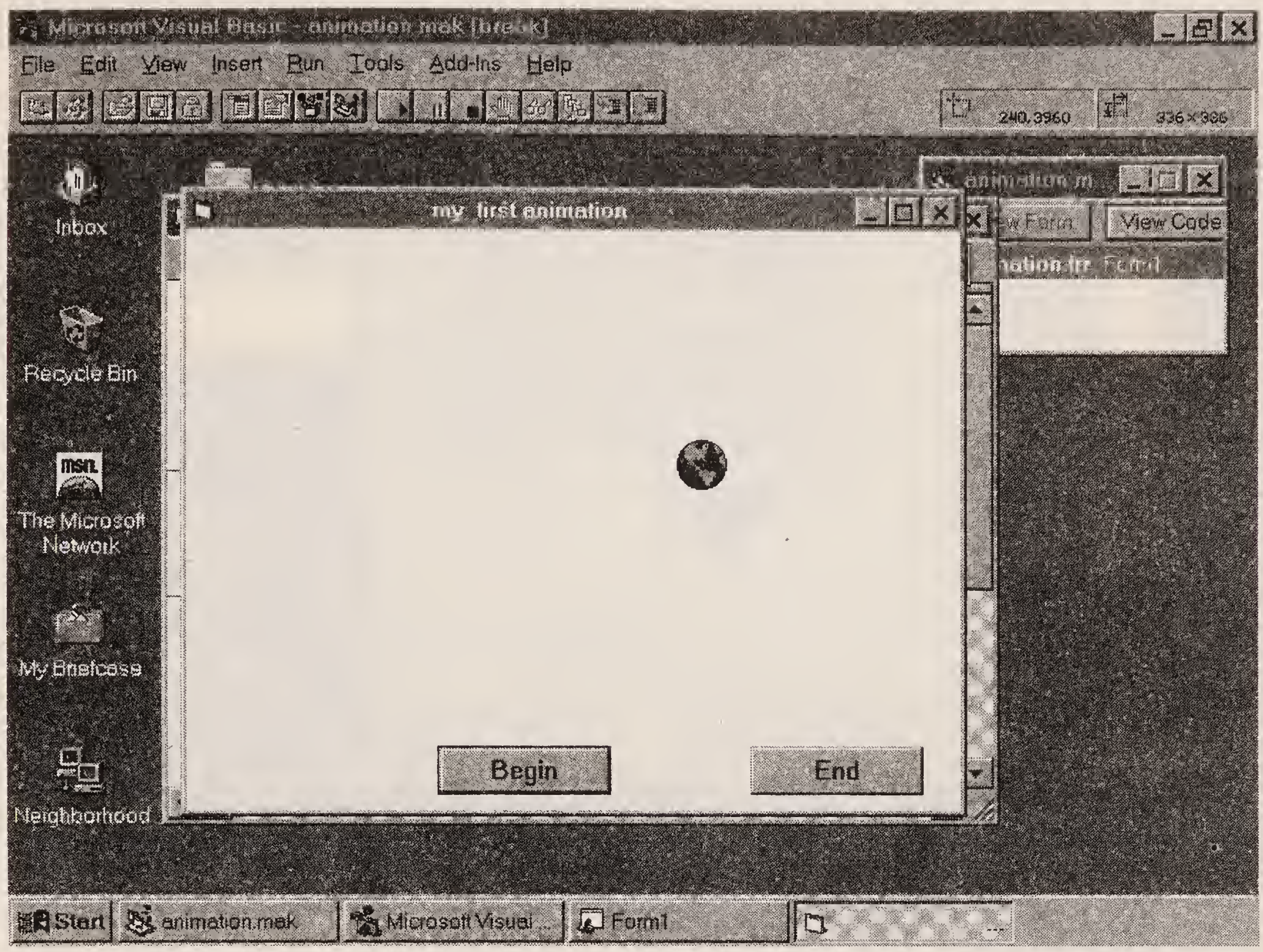

Microsoft Visual Basic - animation.mak [break]
File Edit View Insert Run Tools Add-Ins Help
240, 3960
336 × 336
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Neighborhood
my first animation
Begin
End
animation.m
View Form
View Code
animation.frm Form1
Start
animation.mak
Microsoft Visual ...
Form1

图9.34　在 Form 窗口内来回运动的地球图标

打开刚才那个动画文件的 Code 窗口，在 General 事件中，加入对变量 dx，dy 的声明，即：

```
Dim dx As Integer,dy As Integer
```

然后在 Timer1_ Timer()事件过程中加入下列代码：

```
Image1. Height=Image1. Height+dy
Image1. Width=Image1. Width+dx
If Image1. Height>=1000 Then dy=-50
If Image1. Width>=1000 Then dx=-50
If Image1. Height<=100 Then dy=50
If Image1. Width<=100 Then dx=50
Image1. Stetch=True
```

加入的这段代码意思是，使图像框具有的 Stretch 属性为 True，即可以拉伸图像框中的图形，然后改变图像框的大小，当其大到一定程度时使其变小，当其小到一定程度时使其变大。

在 Form _ Load()事件过程中为 dx，dy 赋值：

```
dx=50;dy=50
```

关闭 Code 窗口，选择 File 菜单中的 Save Project 命令保存文件。

单击工具栏中的 Start 按钮，运行程序。单击 Form 窗口中的 Begin 按钮，即看见一忽大忽小的地球在 Form 窗口内来回运动，就像在飞船中接近和远离地球时看地球一样，读者不妨试一下。如图9.35所示。

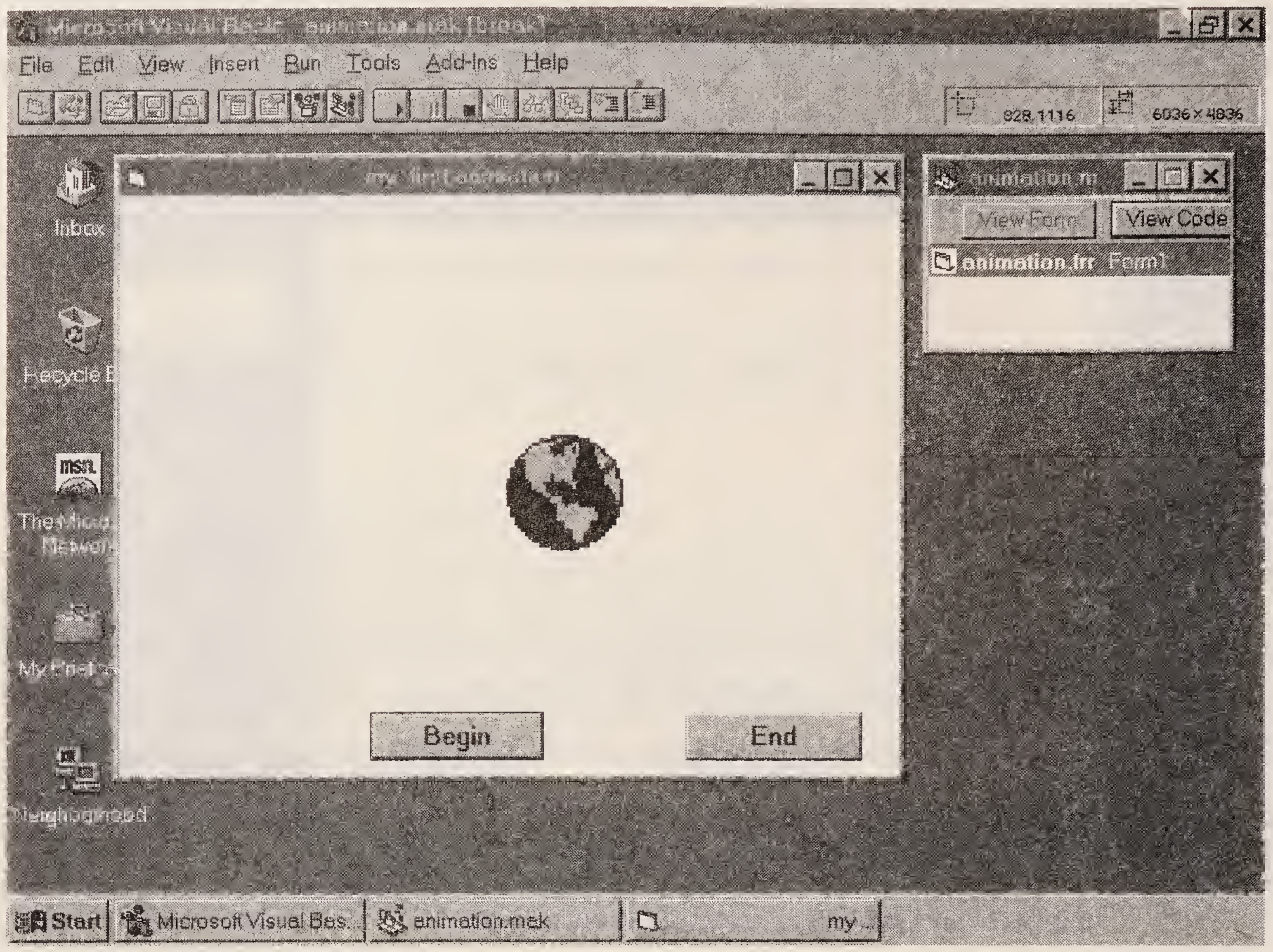

File
Edit
View
Insert
Run
Tools
Add-Ins
Help
928,1116
6036×4836
Inbox
View Form
View Code
animation.frr
Form1
Begin
End
Start
Microsoft Visual Bas...
animation.mak
my...

图9.35　在 Form 窗口中改变体积的运行地球图形

9.8　在 Form 窗口中添加背景图形

最后，我们再来谈谈关于在 Form 窗口中添加背景图形的问题。一般来说，用户只是在 Form 窗口中添加控件或绘制图形。同样的，也可以从元文件或图标等中添加图形到 Form 窗口中作为背景图形。

用户可以通过 Form 的 Picture 属性将图形文件装入 Form 窗口，装入的图形可以是位图、图标或元文件图形。此后，图形将作为 Form 窗口的背景显示在其他控件之后。例如，选择 File 菜单的 New Project 命令，选择 Form 窗口的 Picture 属性，在打开的 Load Picture 对话框中选择 C:\vb\metafile\business\computer.bmp，单击 OK 按钮，则在 Form 窗口中出现一计算机图形，如图9.36所示。

再从工具框中选择图片框工具，在 Form 窗口画一图片框控件，则这一控件覆盖它所包括的那一部分区域。为了使画面美观，我们将这一图片框控件移到 Form 窗口中的计算机图形中的显示屏部分。然后在图片框的 Properties 窗口中选择 Picture 属性，在 Load Picture 对话框中选择 C:\VB\METAFILE\BUSINESS\COMPUTER.BMP，单击 OK 按钮，即在图片框中又出现一与 Form 窗口中的相同的计算机图形。我们在图片框中的计算机图形的屏幕部分再画一图片框控件，装入同一计算机图形，这样一层套一层，形成如图9.37所示图形。

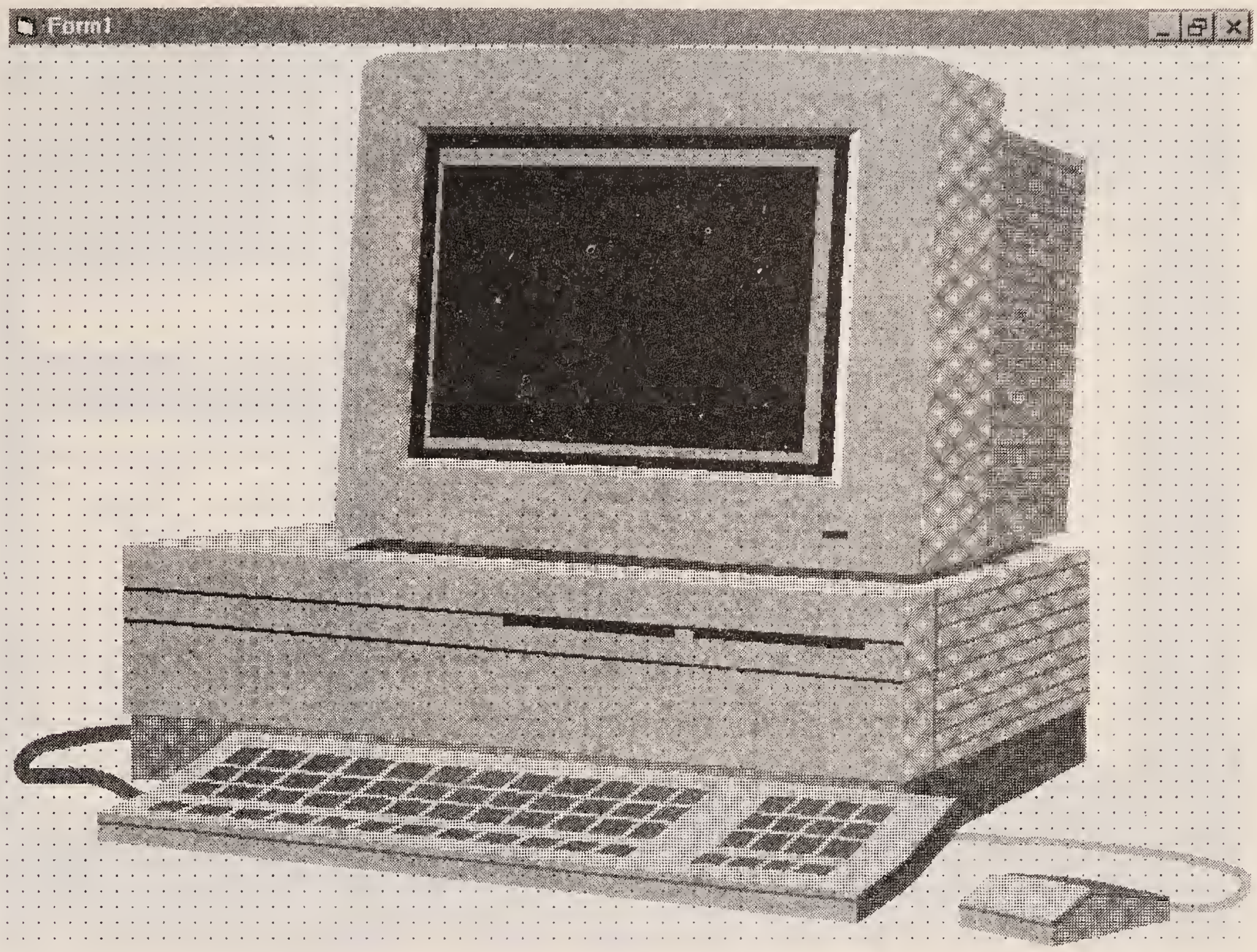


图9.36　Form 窗口中的计算机图形

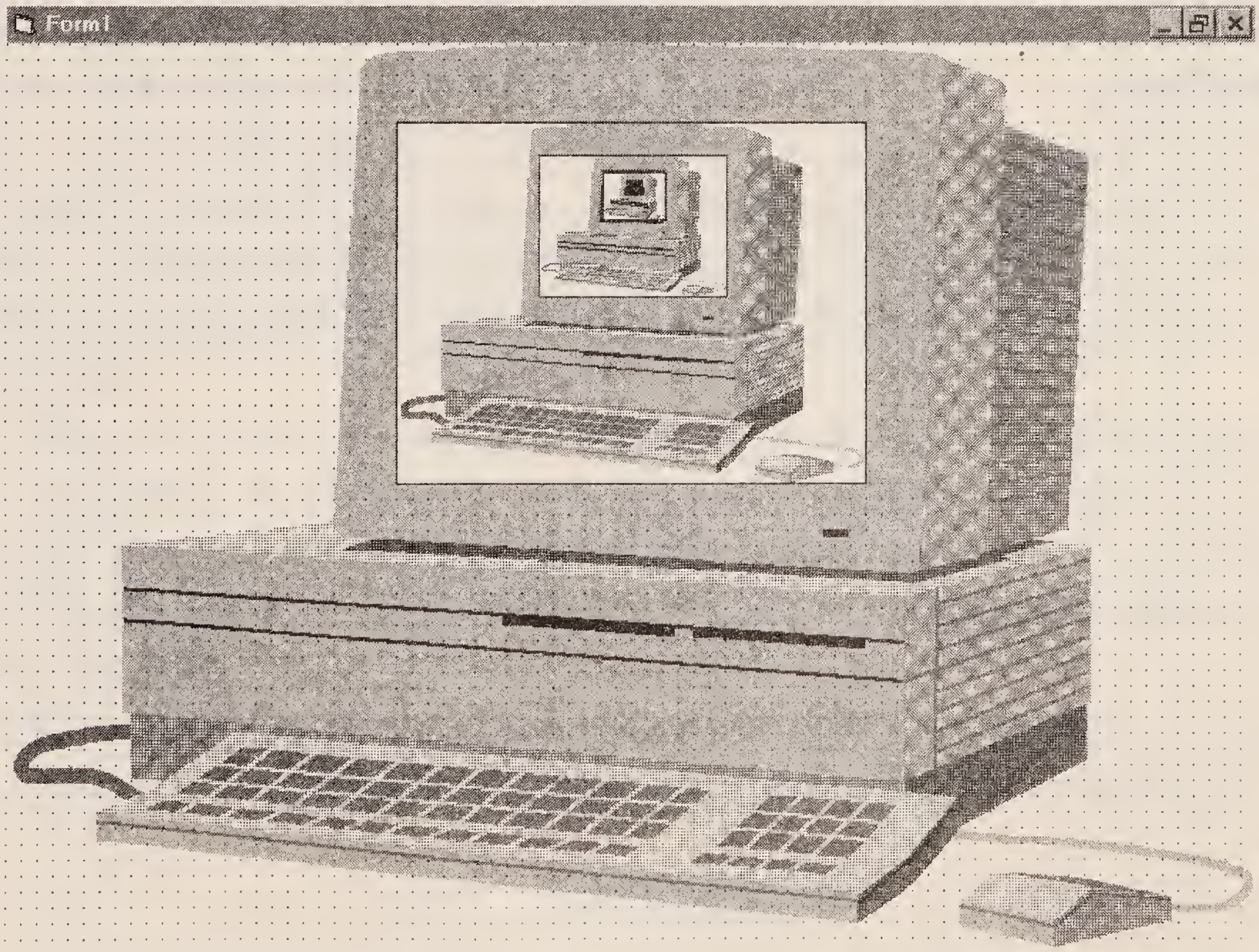
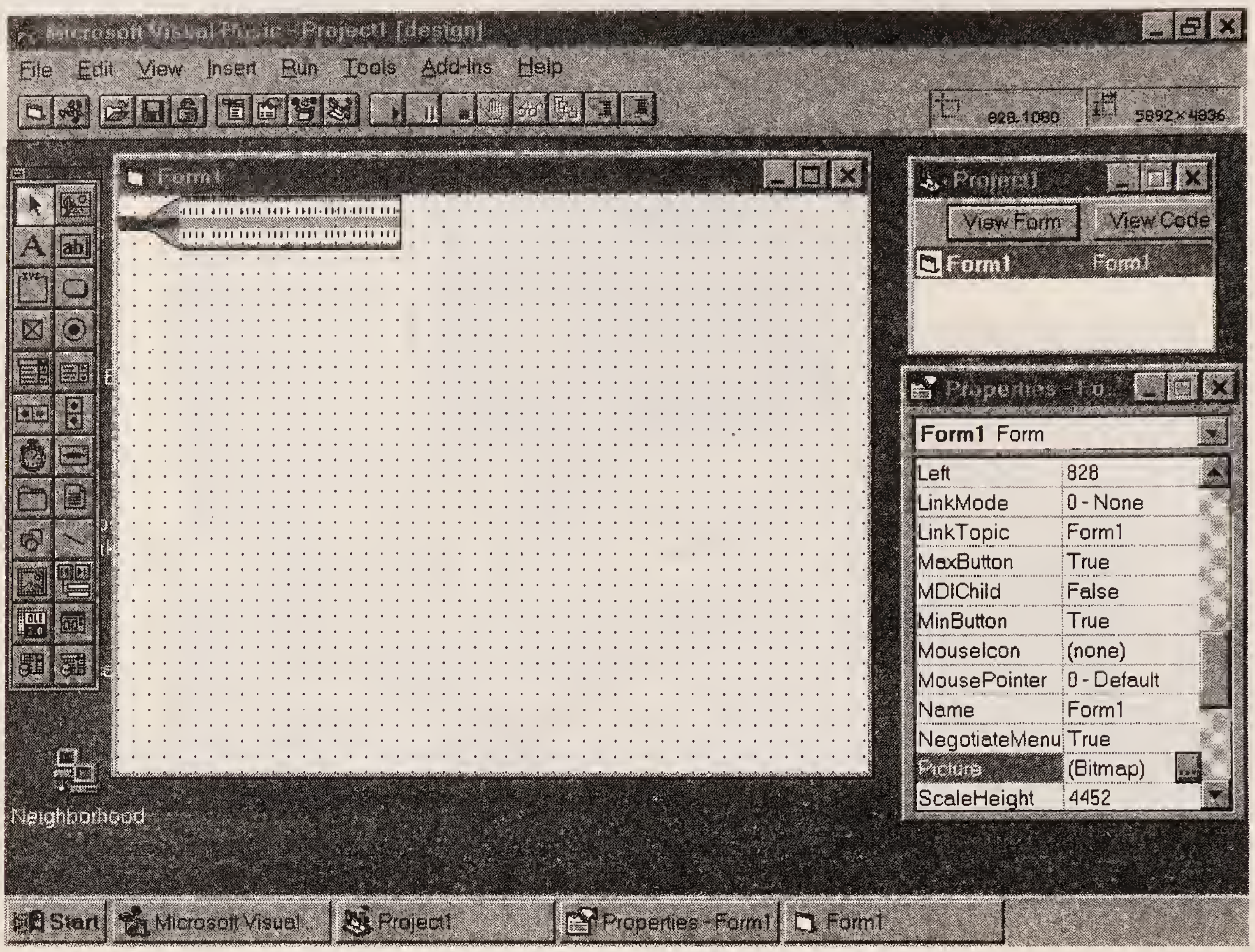


图9.37　Form 窗口中图片框中的嵌套图形

如果将 Form 的 Picture 属性赋值为一元文件图形，则这一图形会自动重设大小以适应 Form 窗口的大小，如图9.36所示。而如果装入的是位图或图标，则它们不会自动重设大小来适应 Form 窗口，且一般都显示在窗体的左上角位置，如图9.38所示。


Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
828,1080
5892×4836
Form1
Project1
View Form
View Code
Form1
Form1
Properties - Fo...
Neighborhood
Start
Microsoft Visual...
Project1
Properties - Form1
Form1

Form1 Form

Left	828
LinkMode	0 - None
LinkTopic	Form1
MaxButton	True
MDIChild	False
MinButton	True
MouseIcon	(none)
MousePointer	0 - Default
Name	Form1
NegotiateMenu	True
Picture	(Bitmap)
ScaleHeight	4452

图9.38　位图或图标一般显示在 Form 窗口的左上角

当然，Form 装载图形时也可以像图片框装载图形一样，在程序代码中通过 LoadPicture()函数来实现。

第十章 文件操作

本章主要讨论了文件的基本操作，包括文件的删除、拷贝、移动和更名。同时分别讨论了有关顺序文件、随机文件和二进制文件的打开、读和写操作的内容。

10.1 文件系统概述

文件可以永久地存储信息，而变量中的值是暂时性的，关闭应用程序时它就会消失。为了长期地保存数据，以便用户再次进行访问和检索，必须将数据存储在文件中。基于成本及使用的考虑，大量数据无法用半导体存储器存储，除了正在执行的程序或极少量的数据调入内存之外，其余大量数据或不执行程序都必须保存外存中，例如磁盘内。外存处理大量数据也带来组织问题，这时我们需要用某种方法来组织数据的存储，以便有效地存取数据，这种组织方法就是文件。由于磁盘存储不在主处理器中，计算机必须执行一定的动作才能把信息放入文件或从文件中取出信息。

Visual Basic 根据计算机访问文件的方式将文件分成三类：顺序文件、随机文件和二进制文件。

顺序文件是普通的正文文件，是最简单的文件结构，它是一个接一个地按序排列，而且只提供第一个记录的存储位置，其它记录的位置无法获悉。所以要找一个数据时，必须从头一个一个地读取直到找到要找的记录为止，顺序文件因此而得名。顺序文件的缺点是：如果要修改数据，必须将所有的数据读入 RAM 中进行修改，然后再将修改好的数据重新写入磁盘。由于无法灵活地随意存取，所以它适用于有规律、不经常修改的数据。优点是所占空间较少，而且容易使用。

随机文件是 Visual Basic 可以按任意次序读写的文件，每一行或每个记录的长度必须相同，在这个文件结构中，每个记录都有一个记录号，所以存入数据时，只要指明第几个记录号，就可以将数据直接存入指定的位置。读取数据时，只要知道记录号，便可以直接读取记录，可以同时作输入与输出，所以随机文件的优点是存取数据快速，更新容易。缺点是所占空间较大，设计程序较烦。

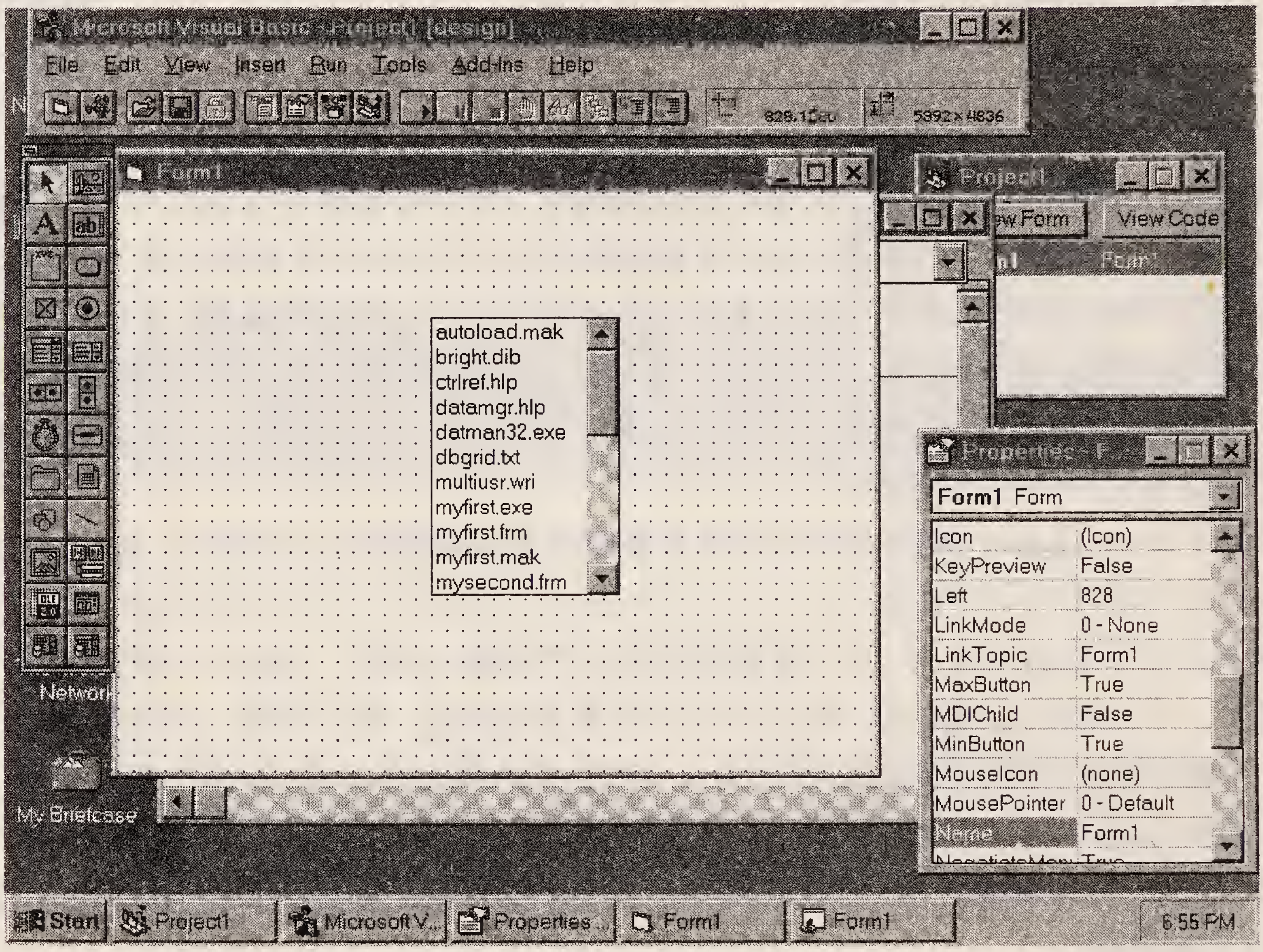
二进制文件是字节的集合，Visual Basic 假定数据不按某种方式进行组织，它允许程序按所需的任何方式组织和访问数据，这类文件的灵活性最大，但程序的工作量也最大。

10.2 文件系统控件

Visual Basic 所提供的文件系统控件允许用户选择新驱动器，观察磁盘的目录结构以及浏览指定目录下的文件。同 Visual Basic 其他控件一样，要使文件系统控件对事件发生响应，用户必须为相应的事件过程编写代码。本节主要讲解文件系统控件的属性及相关的事件，最后讲到怎样使文件系统控件协同工作，以发挥更强大的功能。

10.2.1　文件列表框

缺省的文件列表框显示出当前目录下的文件，见图10.1。可以在设计时或通过代码控制它的一些属性，可使用文件列表框选择文件属性来显示相应的文件名。这对于在应用程序中建立一个允许用户选择一个或一组文件对话十分有用。当列表项数目超过文件列表框的尺寸允许时，Visual Basic 会自动地添加垂直滚动条，用户可用滚动条来浏览文件。用户还必须牢记一点，Windows 习惯于通过双击来选择文件而非单击，这一点对于文件列表框特别重要。文件列表框经常要用到以下属性。


Microsoft Visual Basic - Project1 [design]
File
Edit
View
Insert
Run
Tools
Add-Ins
Help
828,1080
5392 x 4836
Form1
Project1
w Form
View Code
Properties - F...
Network
My Briefcase
Start
Project1
Microsoft V...
Properties...
Form1
Form1
6:55 PM

autoload.mak
bright.dib
ctrlref.hlp
datamgr.hlp
datman32.exe
dbgrid.txt
multiusr.wri
myfirst.exe
myfirst.frm
myfirst.mak
mysecond.frm

Form1 Form

Icon	(Icon)
KeyPreview	False
Left	828
LinkMode	0 - None
LinkTopic	Form1
MaxButton	True
MDIChild	False
MinButton	True
MouseIcon	(none)
MousePointer	0 - Default
Name	Form1

图10.1　文件列表框

(1) Pattern 属性

该属性在运行时确定一个文件列表框中显示的文件名，它接受 DOS 通配符，Pattern 属性值是一个表明文件说明的字符串，诸如“*.*”或“*.FRM”，缺省值为“*.*”，它返回一张包含所有文件的列表。在设计一个应用程序的文件浏览和管理功能时 Pattern 属性起着关键的作用。和其他文件控件属性组合使用 Pattern，可给用户提供多种寻找文件或相似文件的途径。改变 Pattern 属性的值时，Visual Basic 寻找属于列表框的 PatternChange 事件过程，如果有就执行它。

(2) Path 属性

该属性用于或返回文件列表框的当前路径，包括驱动器名，诸如：“C:\VB”或“C:\WIN95\SYSTEM”。Path 属性的缺省路径为当前路径，当建立一个程序的文件浏览和管理功能时可用这个属性。当用户改变 Path 属性时会产生一个 PathChange 事件。

(3) FileName 属性

该属性在运行时确定文件列表框中列表部分被选中的文件。当文件列表框在运行中建

立时，FileName 属性设置成空字符串，意味着当前没有文件被选中。读取此属性时，返回从列表中当前选定的文件。检索改变此属性值可能会引起一个或多个下述事件：PathChange（若改变了路径），PatternChange（若改变了模式）或 Dblclick（若赋值一个已存在的文件名）事件。

（4）List、ListCount 和 ListIndex 属性

这些属性都是用来分析文件列表框所包含的信息。用户可以用 List 属性访问列表中的项，例如第一项为 List(0)，第二项为 List(1)……等等。ListCount 属性决定匹配 Pattern 属性当前目录的文件数。ListIndex 确定文件列表框中当前被选项的索引，该属性在设计时不可用。

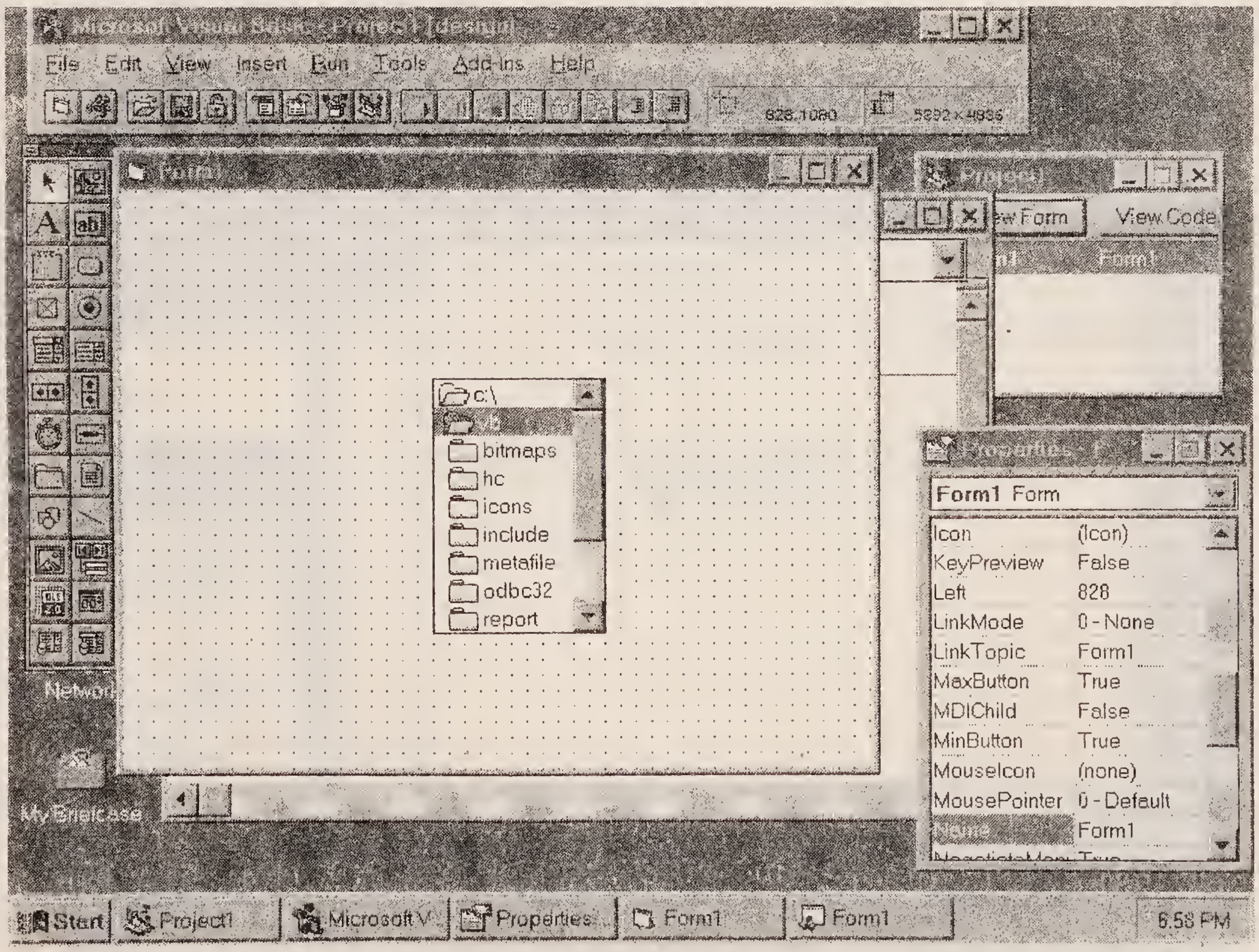
（5）Archive、Hidden、Normal、Readonly 和 System 属性

这些属性控制文件列表框中显示的文件类型，它们的设置值都是有两种—— True 和 False。如果把它们的属性设置为 True，则 Archive 代表文档型文件，Hidden 代表隐含类型文件，Normal 代表普通文件，System 代表系统文件，Readonly 代表只读类型文件。Archive 和 Normal 缺省设置为 True，Hidden、System 和 Readonly 缺省设置为 False。

同文件列表框相关的事件常见的有 Dblclick、Pathchange 和 Patternchange 事件。

10.2.2　目录列表框

目录列表框列出当前驱动器下的目录结构。当前目录的旁边有一个打开的文件夹。当前目录的子目录则标以合上的文件夹，而当前目录的父目录则标以不带光标的打开的文件夹，见图10.2。如果想允许用户在所有可用的目录的文件列表中打开文件对话框，目录列表框将十分有用。


File
Edit
View
Insert
Run
Tools
Add-Ins
Help
828,1080
5892 x 4935
Form1
Project1
View Form
View Code
c:\
vb
bitmaps
hc
icons
include
metafile
odbc32
report
Properties - F
Network
My Briefcase
Start
Project1
Microsoft V
Properties
Form1
Form1
6:58 PM

Form1 Form

Icon	(Icon)
KeyPreview	False
Left	828
LinkMode	0 - None
LinkTopic	Form1
MaxButton	True
MDIChild	False
MinButton	True
MouseIcon	(none)
MousePointer	0 - Default
Name	Form1

图10.2　目录列表框

目录列表框的大部分属性同文件列表框基本一样，目录列表框的 List 属性的功能与文件列表框略有不同。根据目录和子目录建立的序号，当前的扩展目录是－1，当前扩展目录的前一个目录序号是当前目录的绝对值加1，前后再加负号。例如，－2是当前扩展目录的父目录，－3是祖父目录。当前目录以后的目录序号是从0到 ListCount－1，ListCount 是当前目录的子目录数。

目录列表框常与 Click、Change 等事件发生响应，当用户改变目录列表框的 Path 属性时，Visual Basic 就会调用目录列表框的 Change 事件过程。例如，假如用户想在改变 DirBox1 列表框之后，Visual Basic 马上会相应地更新文件列表框，完成这一任务，只需调用如下事件过程：

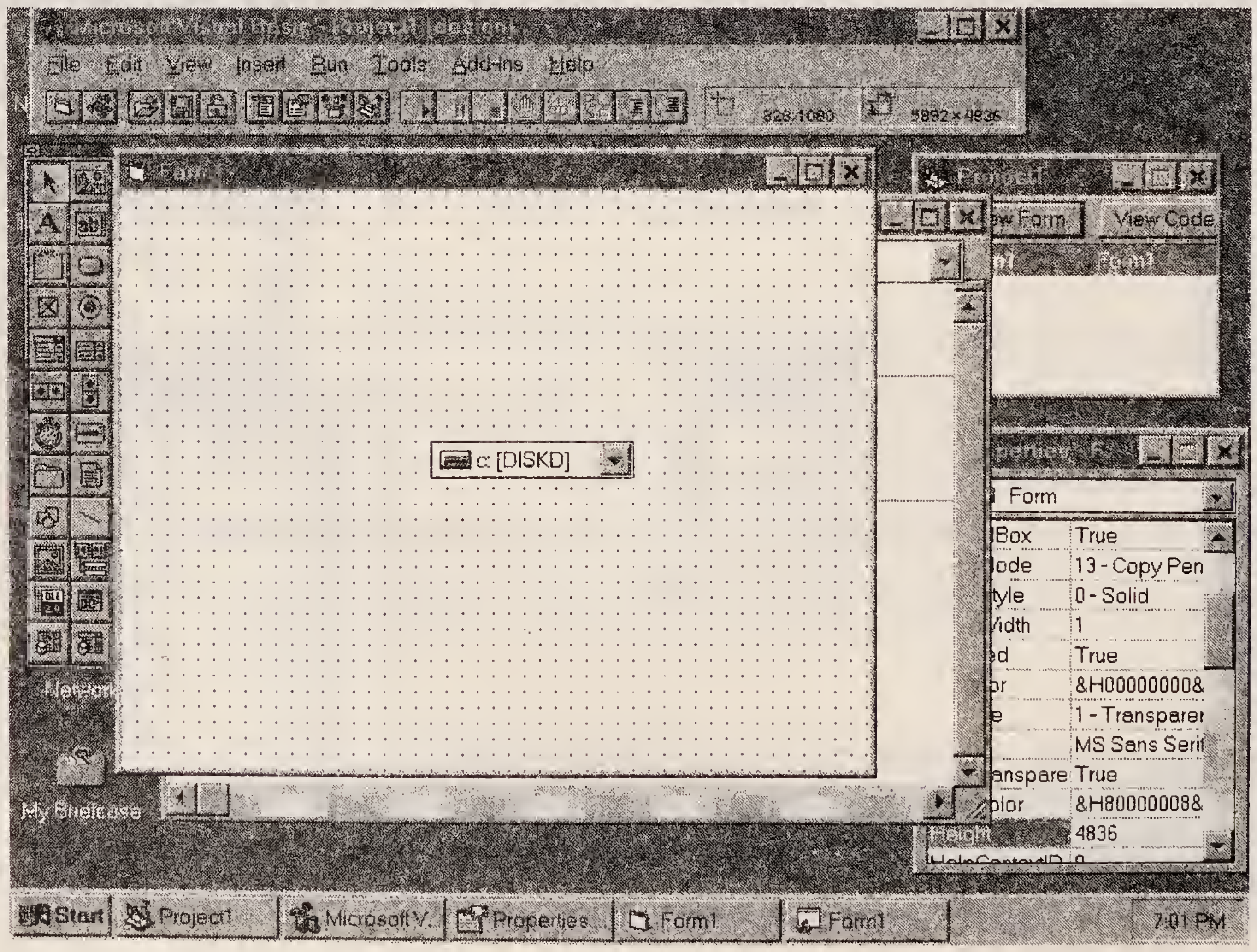
```
Sub DirBox1_ Change()
    File1. Path＝DirBox1. Path
End Sub
```

为了激活此事件过程，用户只需改变目录列表框 Path 属性，可以通过双击列表框的新目录来实现。目录列表框并不认可双击事件，取而代之，它调用 Change 过程响应双击鼠标，并重设 Path 属性。

最后用户应该注意一点，虽然文件列表框和目录列表框都有 Path 属性，但它们并非相同。对于目录列表框，Path 属性指明选择的目录，而对文件列表框，Path 属性指明要到哪里去寻找要显示的文件。

10.2.3　驱动器列表框

不同于文件列表框和目录列表框，驱动器列表框是下拉式的，开始时只显示当前驱动器，当用户单击右边的箭头打开列表框时，驱动器列表框就会显示有效驱动器的列表，供用户选择不同的驱动器，见图10.3。


File
Edit
View
Insert
Run
Tools
Add-ins
Help
Form1
c: [DISKD]
Project1
View Form
View Code
Form
Network
My Briefcase
Start
Project1
Microsoft V...
Properties
Form1
Form1
7:01 PM

Box　True
ode　13 - Copy Pen
tyle　0 - Solid
Vidth　1
d　True
or　&H00000000&
e　1 - Transparent
MS Sans Serif
anspare　True
olor　&H80000008&
Height　4836

图10.3　驱动器列表框

驱动器列表框除了有 List 和 ListIndex 等属性以外，还有一个最重要的 Drive 属性，它确定运行时被选中的驱动器，在设计时不可用。Drive 属性有效范围包括所有的现有的驱动器。

驱动器列表框最常见的相关事件是Change 事件，当用户选择一新驱动器或用代码改变 Drive 属性时发生此事件。

10.2.4 协同文件系统控件

用户在操作中也许希望同时使用上述三种控件，而使这三个控件一起工作非常简单。前面我们讲过与目录列表框和驱动器列表框相关的Change 事件，用户可以为驱动器列表框的Change 事件建立一个过程，当用户选择新的驱动器时就会执行这个事件过程，这个过程将改变目录列表框的路径以符合新选择的驱动器。用户还可以为目录列表框的Change 事件建立一过程，当用户选择新的目录时激发该事件，并改变文件列表框以符合新的目录路径。通过上述的两个过程，三个文件控件就可以同步工作。另外，要对所选择的文件有所动作，用户仍然需要编写另外的过程。下面我们将以一个例子为主，讲解如何编写代码使三个文件控件同步工作，并输出被选择文件的路径名。

(1) 建立初始化界面

在窗体上加上目录列表框、文件列表框、驱动器列表框及命令按钮等控件，并利用属性窗口设置控件属性如下表，建立如图10.4所示的初始化界面。

控件	Name	Caption	Height	Width
文件列表框	File1	没有	1752	1452
目录列表框	Dir1	没有	1752	1452
驱动器列表框	Drive1	没有	288	972
命令按钮1	BtnSelect	Select	372	972
命令按钮2	BtnCancel	Cancel	372	972
命令按钮3	BtnMessage	Message	372	972

注意　驱动器列表框的Height 属性在设计时不能修改，它为常数288。

现在用户单击驱动器列表框或者目录列表框时都会发现文件列表框没有什么反应，这是因为这三个控件还没有能够协同工作，下面我们就编写代码使它们能够相互传递信息。

(2) 编写代码

用户双击窗体的空白处，打开代码窗口，在相应的事件过程编写以下代码：

```
Sub Dir1_Change()
    File1.Path=Dir1.Path
End Sub
Sub Drive1_Change()
    Dir1.Path=Drive1.Drive
End Sub
```

第一个过程设置了 File 控件的 Path 属性以符合 Dir1控件的 Path 属性，第二个过程设置 Dir1的 Path 属性以符合 Drive1控件的 Drive 属性，当用户选择不同的驱动器时，Dir1控件的路径就相应地发生变化。Dir1控件路径的变化又赋给 File1控件的 Path 属性，从而实现文件系统控件的协同工作。下面的代码作用是显示被选择文件的路径名：

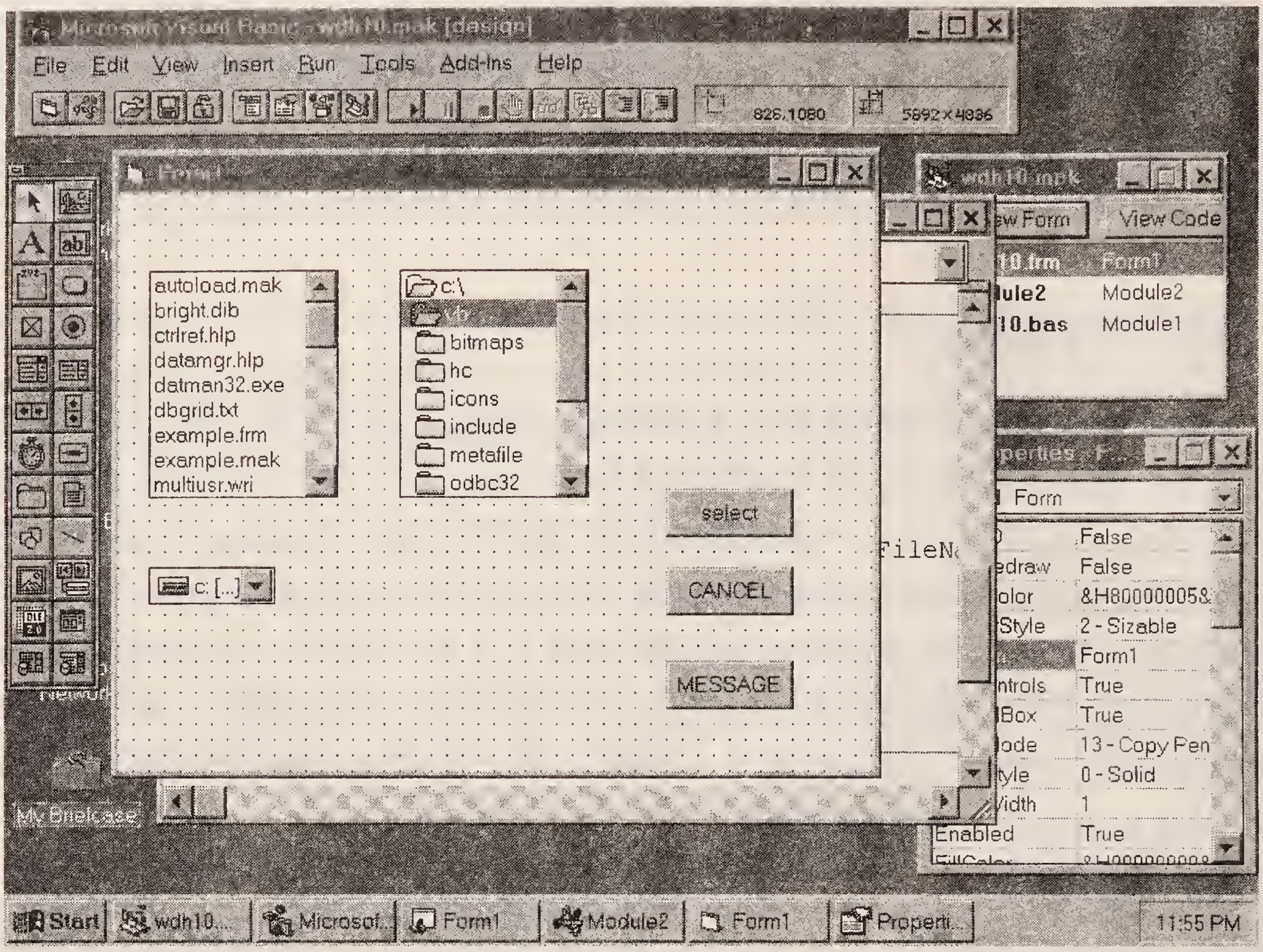
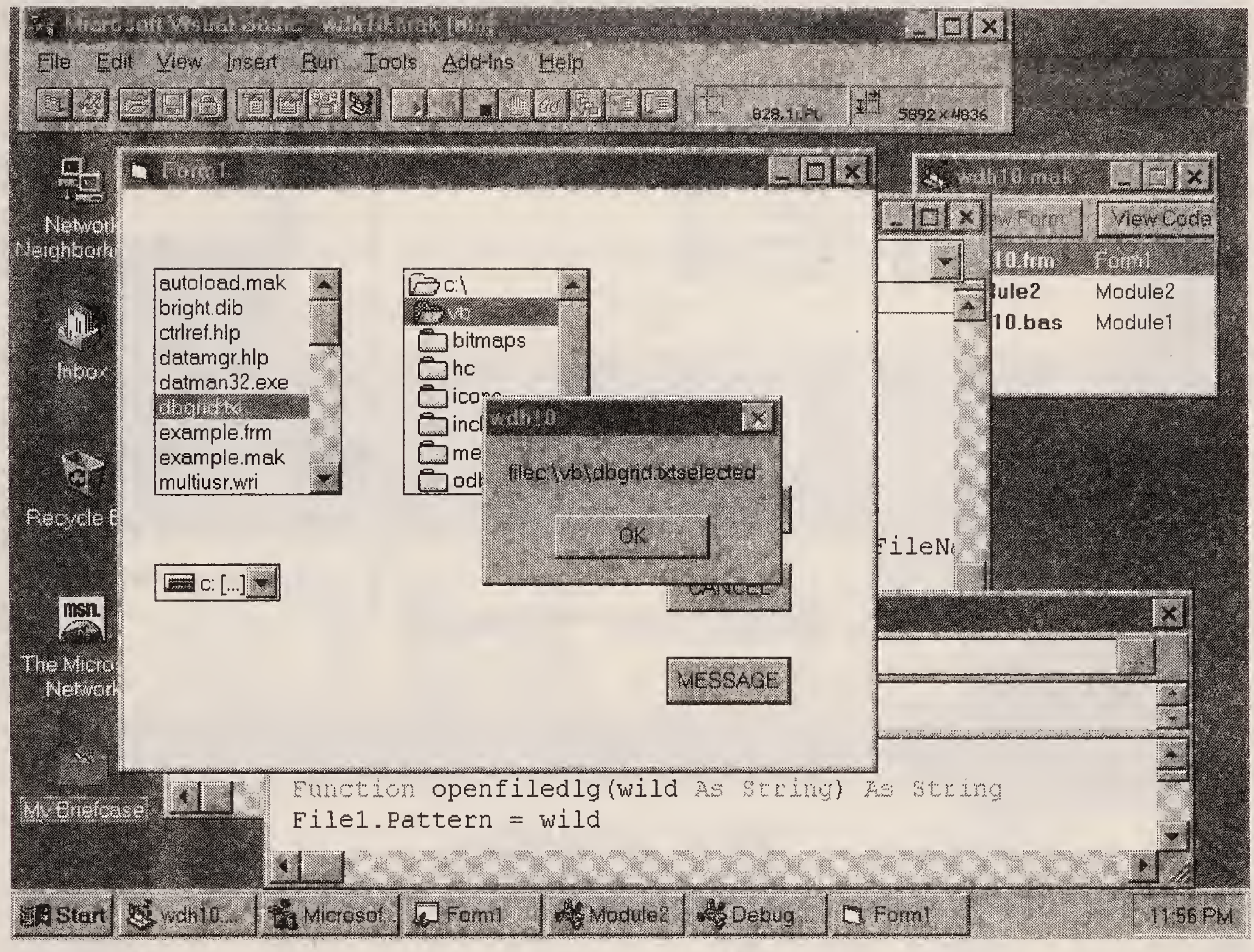

File
Edit
View
Insert
Run
Tools
Add-Ins
Help
826,1080
5892×4836
Form1
autoload.mak
bright.dib
ctrlref.hlp
datamgr.hlp
datman32.exe
dbgrid.txt
example.frm
example.mak
multiusr.wri
c:\
vb
bitmaps
hc
icons
include
metafile
odbc32
c: [...]
select
CANCEL
MESSAGE
FileN
wdh10.mak
View Form
View Code
10.frm Form1
Module2 Module2
10.bas Module1
Properties - F...
Form
False
False
&H80000005&
2 - Sizable
Form1
True
True
13 - Copy Pen
0 - Solid
1
Enabled True
My Briefcase
Start
wdh10...
Microsof...
Form1
Module2
Form1
Properti...
11:55 PM

图10.4　初始化界面

```
Dim SelectedFile As string
Function OpenFileDlg (Wild As String) As String
    File1. Pattern=Wild
    OpenFileDlg=SelectedFile
End Function

Sub BtnMessage _ click()
    Selected=OpenFileDlg("*.*")
    If Selected=" "Then
        MsgBox "Cancel"
    Else
        MsgBox "File" & Selected & "selected"
    End If
End Sub

Sub BtnSelect _ Click()
    If File1. ListIndex>=0 Then
        SelectedFile=File1. Path
    If Right(SelectedFile,1)<>"\"Then
        SelectedFile=SelectedFile & "\"
    End If
        SelectedFile=SelectedFile & File1. FileName
    End If
End Sub
```

用户按 F5键运行程序，选择一个文件，单击 Select 按钮，然后单击 Message 按钮显示被选择文件的路径信息，见图10.5。


Microsoft Visual Basic - wdh10.mak [run]
File
Edit
View
Insert
Run
Tools
Add-Ins
Help
828.1 Pt
5892 x 4836
Network Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Form1
autoload.mak
bright.dib
ctrlref.hlp
datamgr.hlp
datman32.exe
dbgrid.txt
example.frm
example.mak
multiusr.wri
c:\
vb
bitmaps
hc
icons
incl
me
odk
c: [...]
wdh10
OK
CANCEL
MESSAGE
wdh10.mak
View Form
View Code
10.frm
Form1
ule2
Module2
10.bas
Module1
FileN
Start
wdh10...
Microsof...
Form1
Module2
Debug...
Form1
11:56 PM

filec:\vb\dbgrid.txtselected

```
Function openfiledlg(wild As String) As String
File1.Pattern = wild
```

图10.5　显示路径信息

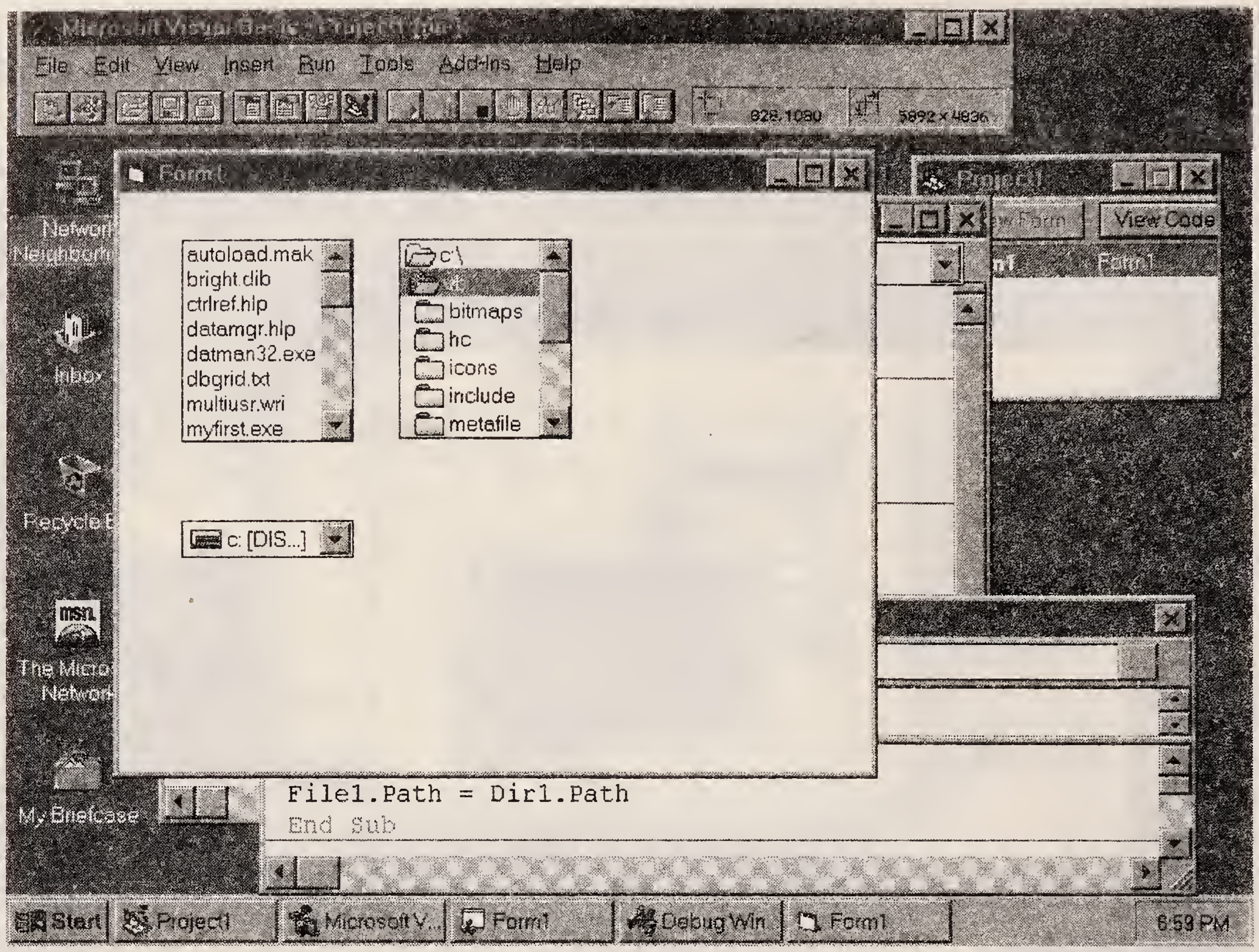
10.3　文件基本操作

上节用户学会了怎样使文件系统控件协同地工作，也许用户想通过双击文件名来运行一个程序或者想删除、拷贝文件，这节我们将以一个例子为线索具体讲解文件的删除、拷贝、移动和更名。

10.3.1　准备工作

为了使用户能够方便地选择文件进行有关操作，我们把三个文件系统控件加到窗体上，见图10.6，并编写以下代码使它们能够协同工作，为有关文件操作做些准备。

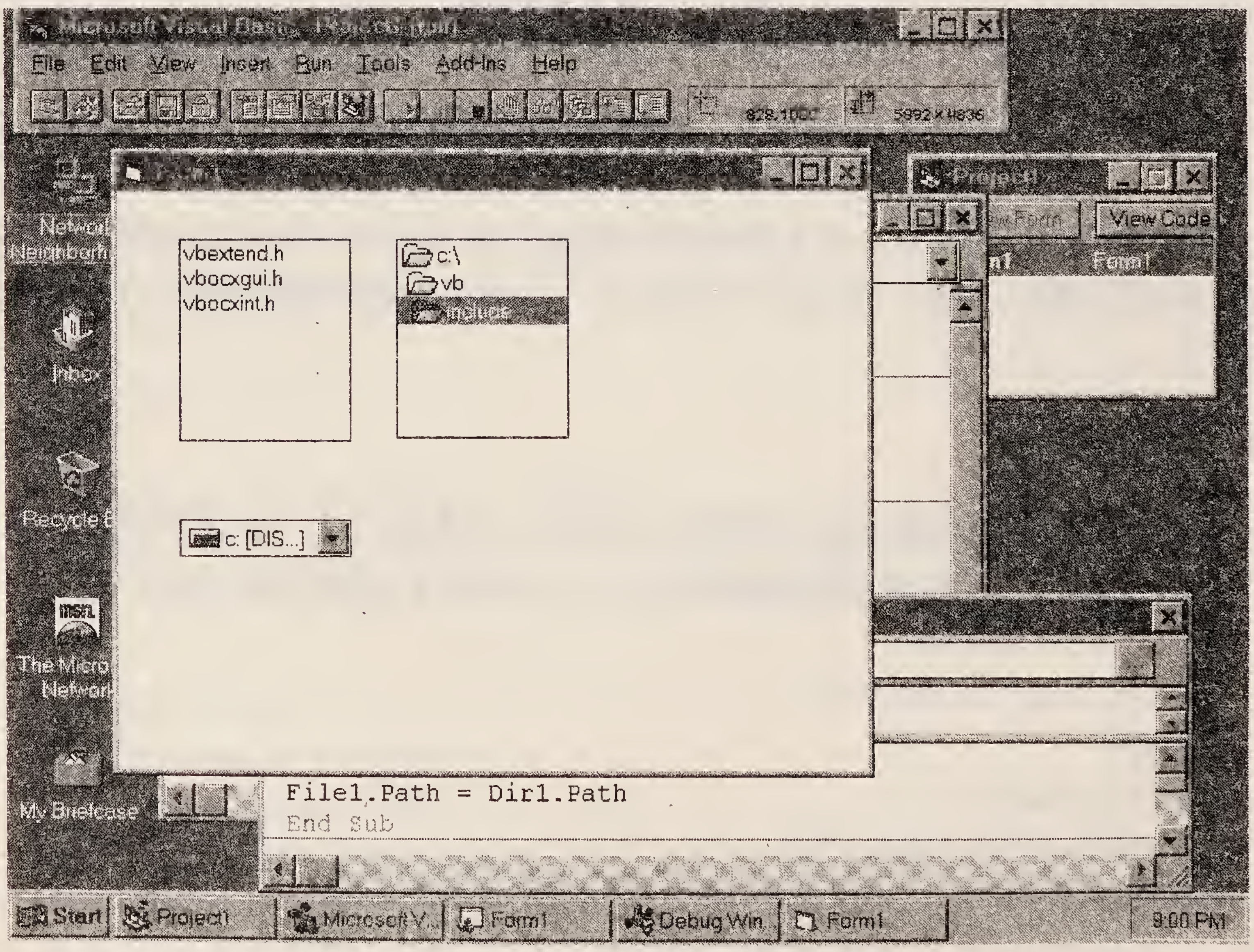
```
Sub Drive 1_ Change()
    Dir1. Path=Drive1. Drive
End Sub
Sub Dir1_ Change()
    FileList1. Path=Drive. Path
End Sub
```


File Edit View Insert Run Tools Add-Ins Help
828,1080
5892 x 4836
Form1
Project1
View Form
View Code
Form1
autoload.mak
bright.dib
ctrlref.hlp
datamgr.hlp
datman32.exe
dbgrid.txt
multiusr.wri
myfirst.exe
c:\
vb
bitmaps
hc
icons
include
metafile
c: [DIS...]
Network Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Start
Project1
Microsoft V...
Form1
Debug Win
Form1
6:59 PM

```
File1.Path = Dir1.Path
End Sub
```

图10.6　文件系统控件

当用户双击目录列表框或单击驱动器列表框时，可以看到文件列表框都有所反应，例如用户双击 INCLUDE 目录时，显示出图10.7，可以看出文件列表框的文件与图10.6不相同。


File Edit View Insert Run Tools Add-Ins Help
828,1080
5892 x 4836
Project1
View Form
View Code
Form1
vbextend.h
vbocxgui.h
vbocxint.h
c:\
vb
include
c: [DIS...]
Network Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Start
Project1
Microsoft V...
Form1
Debug Win
Form1
9:00 PM

```
File1.Path = Dir1.Path
End Sub
```

图10.7　INCLUDE 目录下的文件显示

把上面两段代码加到相应的事件过程中就能够协同文件系统的控件，用户可以选择不同驱动器不同目录下的文件进行有关操作。为了以后的使用，用户可以用 EXAMPLE. FRM、EXAMPLE. MAK 保存该项目文件。

10.3.2 文件的删除

在 Visual Basic 程序中删除一个文件很简单，使用 kill 语句并为它提供欲删除的文件路径名即可删除一个文件。

用户可以打开 EXAMPLE. MAK 文件并在窗体加上一个命令按钮。并设置它的 Name 属性为 CmRemove，Caption 属性为 Remove，见图10.8。

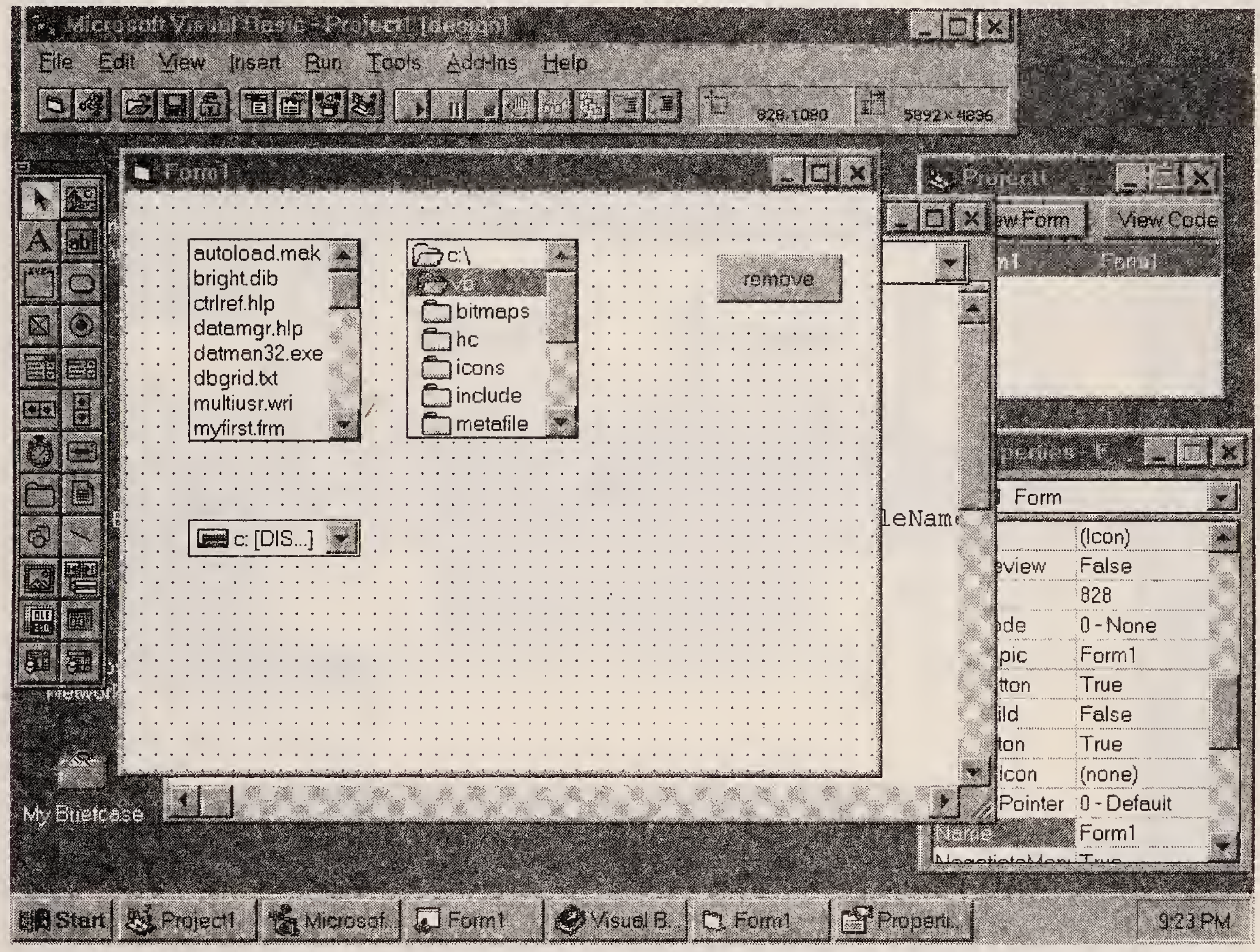

Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
828,1080
5892 x 4896
Form1
autoload.mak
bright.dib
ctrlref.hlp
datamgr.hlp
datman32.exe
dbgrid.txt
multiusr.wri
myfirst.frm
c:\
vb
bitmaps
hc
icons
include
metafile
remove
c: [DIS...]
Project1
View Form
View Code
Form
(Icon)
False
828
0 - None
Form1
True
False
True
(none)
0 - Default
Form1
My Briefcase
Start
Project1
Microsof...
Form1
Visual B...
Form1
Properti...
9:23 PM

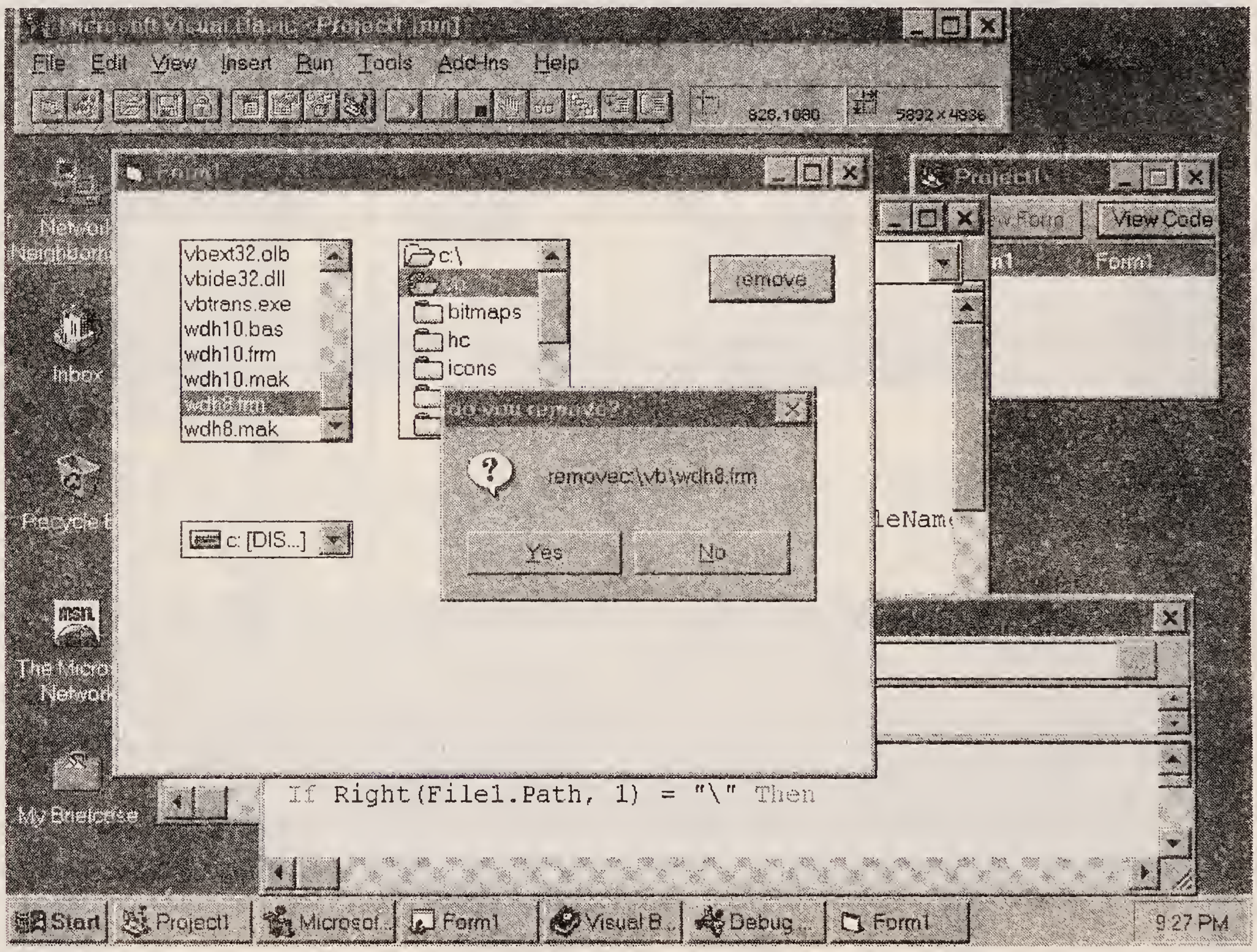
图10.8 加上删除按钮

用户双击 remove 按钮打开代码窗口，在 cmremove _ click()过程中加入以下代码：

```
Sub CmRemove _ click()
    If Right(FileList1. Path)<>"\" Then
        FileRemove=FileList1. Path & "\" & FileList1. FileName
    Else
        FileRemove=FileList1. Path & FileList1. FileName
    End If
        Msg1 $ ="Remove" & FileRemove
        Msg2 $ ="Do you remove?"
        retval=MsgBox(Msg1 $ ,36,Msg2 $ )
    If retval=6 Then
        kill FileRemove
        FileList1. Refresh
        End If
    End Sub
```

在程序的开始加了一段有关测试目录的条件。在一般情况下，只有文件存在于根目录的情况下，才能简单地将 Path 和 FileName 属性连结起来以获取文件名。因为这时 Path 属性是以"\"结尾，因此对于不在根目录下的文件，用户必须加上反斜杠以正确获取文件名。程序的最后用到了 Refresh 方法，该方法强制性地将文件列表框的对象更新，当文件列表框或者是目录列表框有变化时，就可用 Refresh 刷新列表。

当用户按 F5键运行程序后，选择想删除的文件名，然后单击 Remove 命令按钮，这时出现一消息框询问用户是否真的删除文件，如果选择 Yes 按钮，则执行删除操作，选择 No 按钮，则取消删除操作，见图10.9。


File
Edit
View
Insert
Run
Tools
Add-Ins
Help
828,1080
5892×4836
Network Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
vbext32.olb
vbide32.dll
vbtrans.exe
wdh10.bas
wdh10.frm
wdh10.mak
wdh8.frm
wdh8.mak
c: [DIS...]
c:\
bitmaps
hc
icons
remove
Yes
No
Project1
View Form
View Code
Form1
Start
Project1
Microsof...
Form1
Visual B...
Debug...
Form1
9:27 PM

removec:\vb\wdh8.frm

```
If Right(File1.Path, 1) = "\" Then
```

图10.9　删除文件消息框

用户运行完程序以后，单击工具条上的 End 图标结束程序运行，并以 Example.mak 把该文件存盘。

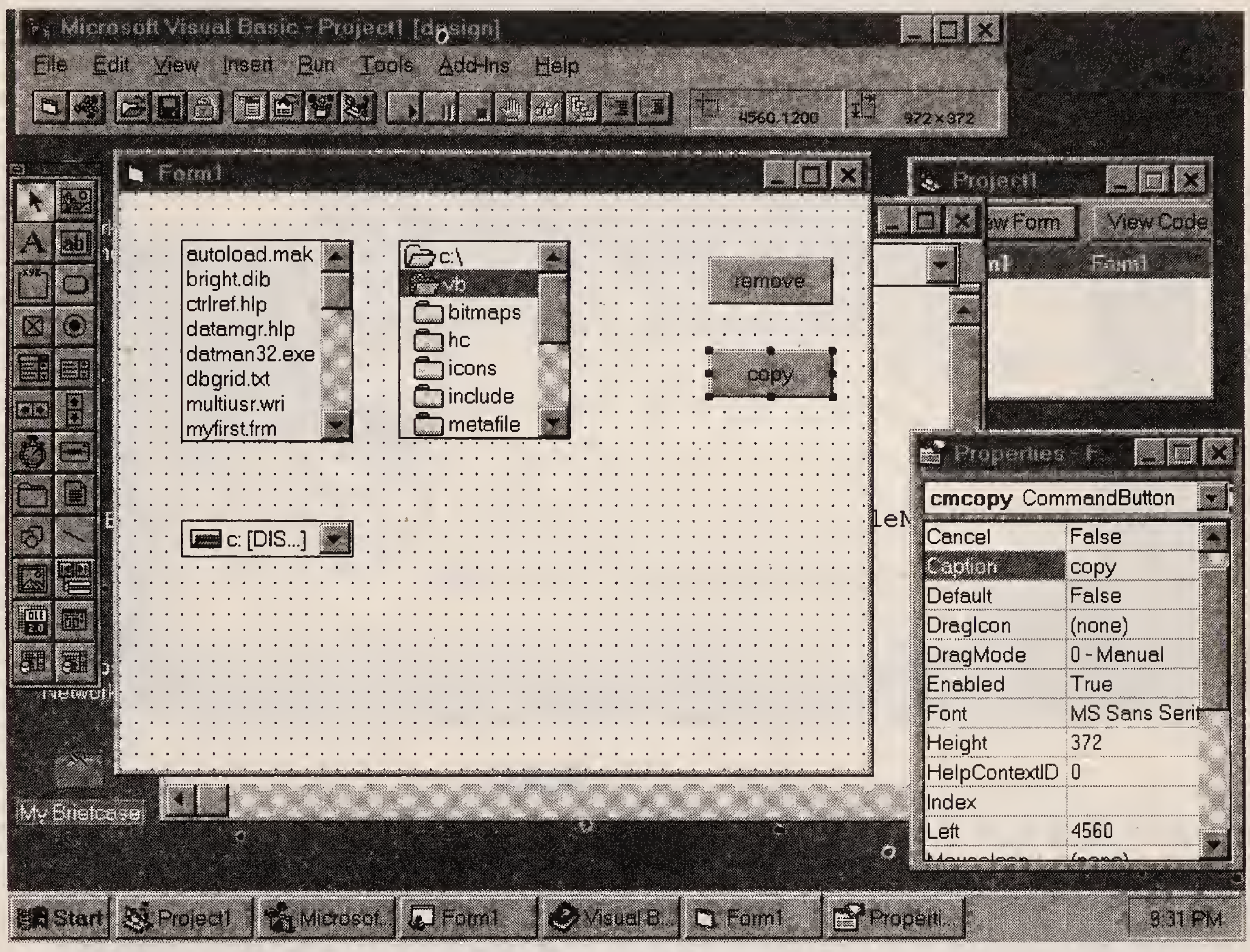
10.3.3　拷贝和移动文件

Visual Basic 提供了一个 FileCopy 语句用于拷贝一个文件，FileCopy 语句的用法格式为：

```
FileCopy srcfile,destfile
```

其中 srcfile 指定被拷贝文件的明确文件名的字符串表达式，不含有匹配符，可以包括驱动器和路径信息。destfile 指定 srcfile 要拷贝到的明确文件名（没有匹配符）的字符串表达式，可以包括驱动器和路径信息。用户要注意的是不能拷贝已由 Visual Basic 打开的文件作只读以外的访问。

用户打开 EXAMPLE. MAK 项目文件，在窗体上加一个命令按钮，并用属性窗口把它的 Name 属性设置为 CmCopy，Caption 属性设置为 Copy，见图10.10。


Microsoft Visual Basic - Project1 [design]
File
Edit
View
Insert
Run
Tools
Add-Ins
Help
4560,1200
972 x 372
Form1
autoload.mak
bright.dib
ctrlref.hlp
datamgr.hlp
datman32.exe
dbgrid.txt
multiusr.wri
myfirst.frm
c:\
vb
bitmaps
hc
icons
include
metafile
c: [DIS...]
remove
copy
Project1
View Form
View Code
Form1
Properties - F
My Briefcase
Start
Project1
Microsof...
Form1
Visual B...
Form1
Properti...
9:31 PM

cmcopy CommandButton

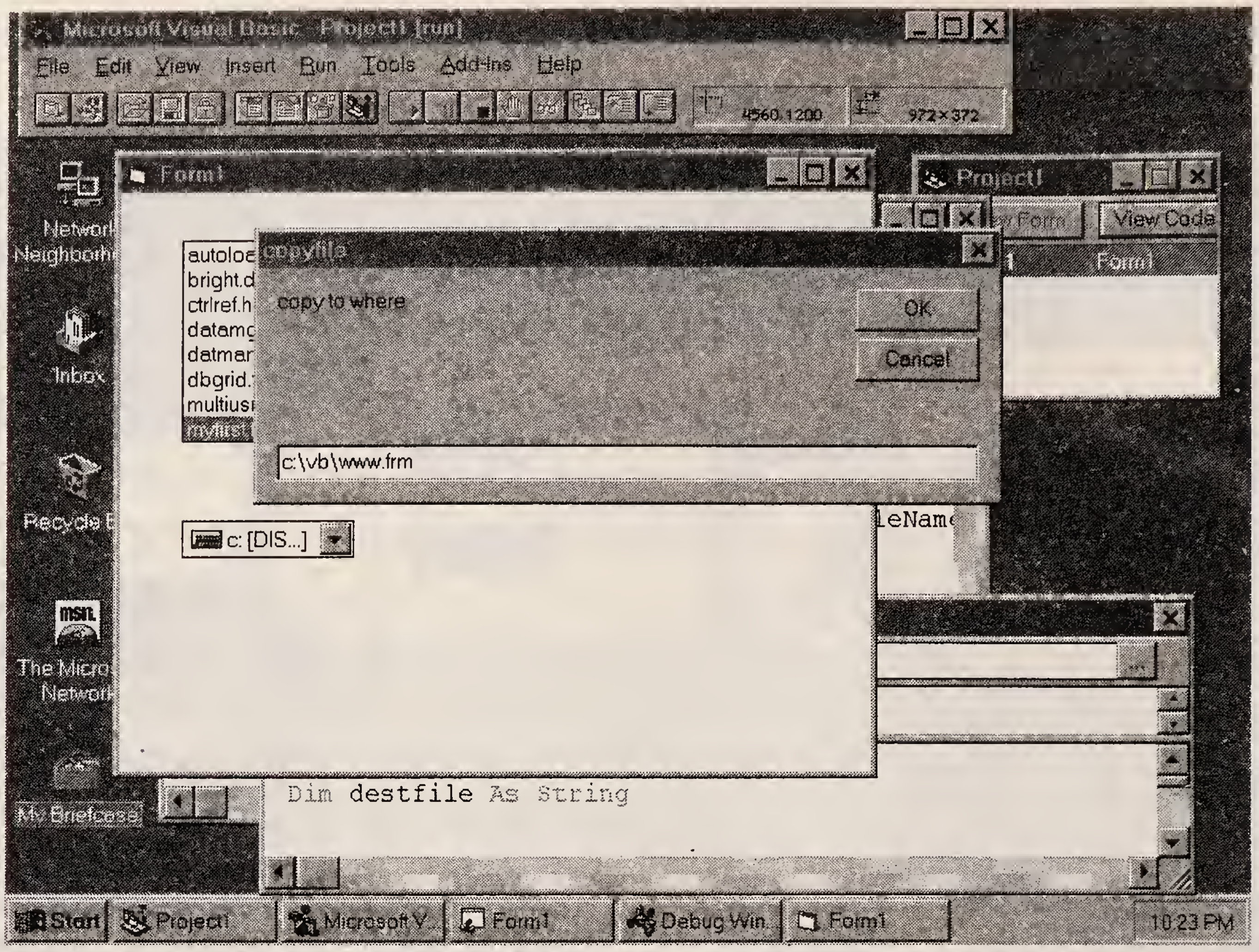
Cancel	False
Caption	copy
Default	False
DragIcon	(none)
DragMode	0 - Manual
Enabled	True
Font	MS Sans Serif
Height	372
HelpContextID	0
Index	
Left	4560

图10.10　增加 copy 按钮

双击 Copy 命令按钮打开代码窗口，在 Sub CmCopy _ click()事件过程中键入以下代码：

```
Sub CmCopy _ click()
    Dim DestFile As String              'Declare rariables
    on Error GoTo ErrHandler
    DestFile=InputBox("Copy to where","copyfile")
    Exit Sub
ErrHandler:
If Err=55 Then                          'File already open
    MsgBox "Cann't copy a open file"
Else
    MsgBox "You must specify a file name"
End If
    Resume Next
End Sub
```

当用户按 F5键运行程序时，在文件列表框单击用户想拷贝的文件名，然后单击 Copy 按钮，这时屏幕出现一输入框提示用户输入放置拷贝文件的路径名，用户可以向输入框中键入完整的路径名。例如 C:\VB\WWW. FRM，见图10.11，单击 OK，这样就把选择的文件拷到该目录下。


Microsoft Visual Basic - Project1 [run]
File Edit View Insert Run Tools Add-Ins Help
4560,1200
972×372
Form1
copyfile
OK
Cancel
c: [DIS...]
Project1
View Form
View Code
Form1
Network Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Start
Project1
Microsoft V...
Form1
Debug Win...
Form1
10:23 PM

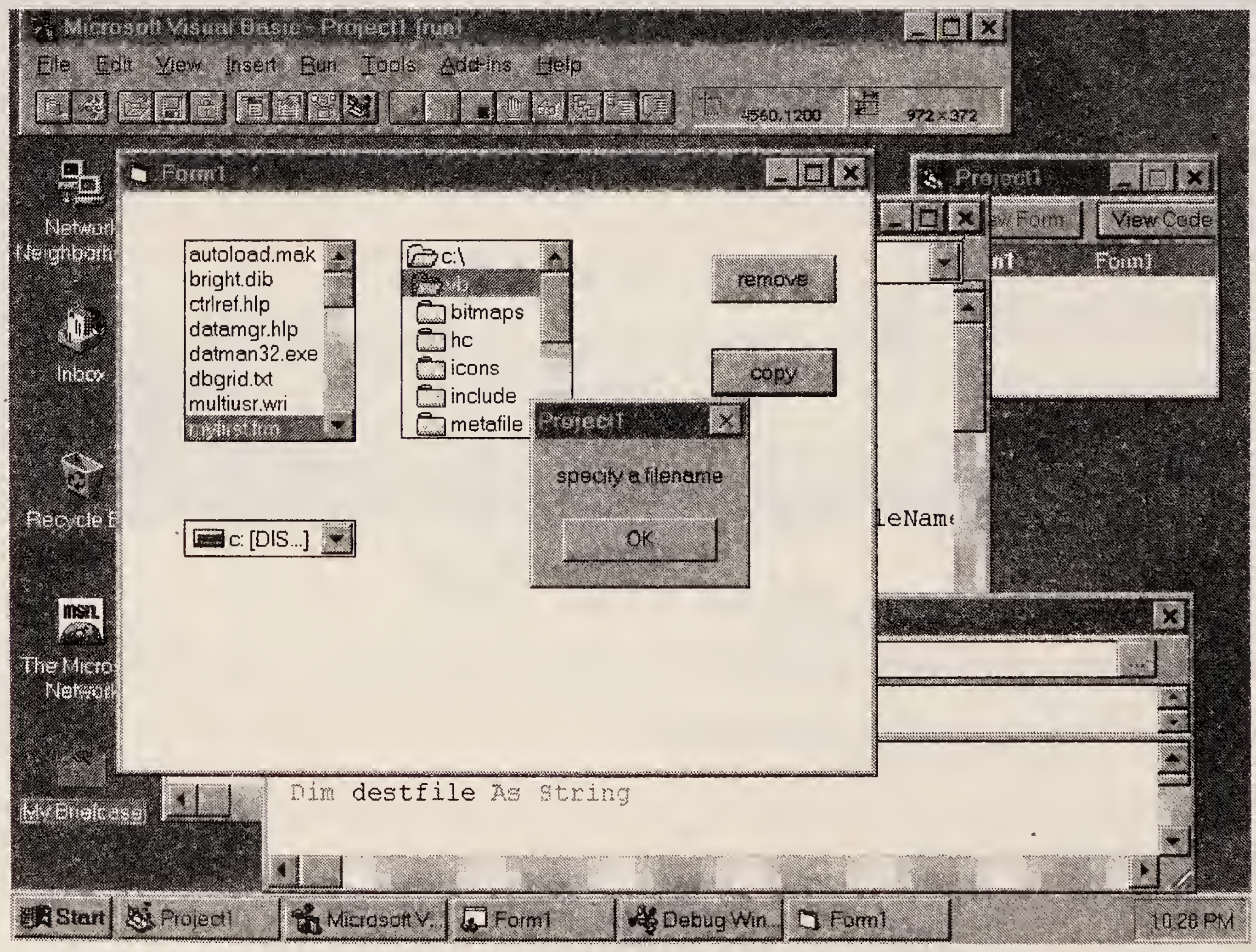
copy to where

c:\vb\www.frm

```
Dim destfile As String
```

图10.11　输入路径名

当用户没有选择文件名而单击 Copy 按钮，然后在输入框中键入路径名再单击 OK 按钮时，屏幕上会出现一消息框提示用户确定一个要拷贝的文件，见图10.12。


Microsoft Visual Basic - Project1 [run]
File Edit View Insert Run Tools Add-Ins Help
4560,1200
972×372
Form1
autoload.mak
bright.dib
ctrlref.hlp
datamgr.hlp
datman32.exe
dbgrid.txt
multiusr.wri
myfirst.frm
c:\
vb
bitmaps
hc
icons
include
metafile
remove
copy
Project1
OK
c: [DIS...]
View Form
View Code
Form1
Network Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Start
Project1
Microsoft V...
Form1
Debug Win...
Form1
10:28 PM

specify a filename

```
Dim destfile As String
```

图10.12　提示选择文件

用户可以单击工具条上的 End 命令图标结束程序运行，并以 EXAMPLE.MAK 保存该项目文件。

现在用户已学会怎样删除和拷贝文件了，其实也同时学会了怎样移动文件。用户可以先拷贝文件，然后删除被拷贝的文件就可实现文件的移动。

10.3.4 文件的更名

在 Visual Basic 中给文件更名很容易，用 Name 语句就可以了。Name 语句的语法格式如下：

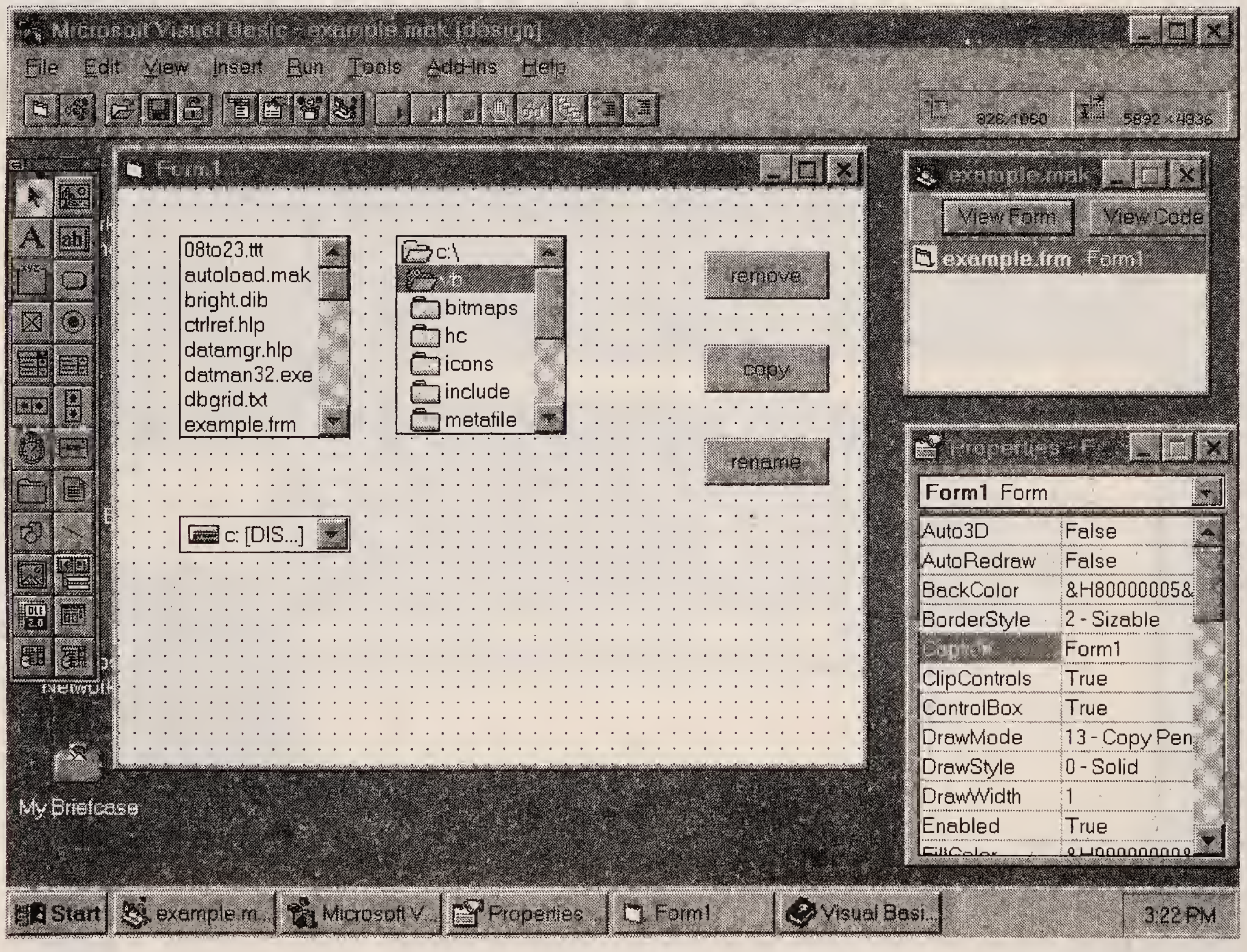
Name OldPathName As NewPathName

其中 OldPathName 是一字符串表达式，指定已存在的文件及路径——可能包括目录或文件夹和驱动器。NewPathName 也是一字符串表达式，它确定新文件名及路径——可能包括目录或文件夹及驱动器，被 NewPathName 指定的文件名不能是已存在的文件名。

NewPathName 和 OldPathName 必须在同一驱动器上。如果 NewPathName 指定的路径存在并且不同于 OldPathName 指定的路径，则 Name 语句移动文件到新的目录或文件夹中并更改文件名。如果 NewPathName 和 OldPathName 指定不同的路径和同一文件名，则 Name 语句移动文件到新的地点且保持文件名不改变。从这里可以看出，Name 语句不仅有更名功能，而且还有移动文件功能。利用 Name 语句，用户从一个目录或文件夹中移动一个文件到另一个目录或文件夹中，但是用户不能够移动一个目录或一个文件夹。

注意　使用 Name 语句对一个已被打开的文件更名会产生一个错误，所以对被打开的文件更名之前必须关闭该文件。

打开 EXAMPLE.MAK 项目文件，在窗体上增加一个命令按钮并设置 Name 属性和 Caption 属性别为 CmRename 和 rename，见图10.13。


Microsoft Visual Basic - example.mak [design]
File Edit View Insert Run Tools Add-Ins Help
826,1060
5892 x 4836
Form1
08to23.ttt
autoload.mak
bright.dib
ctrlref.hlp
datamgr.hlp
datman32.exe
dbgrid.txt
example.frm
c:\
vb
bitmaps
hc
icons
include
metafile
c: [DIS...]
remove
copy
rename
example.mak
View Form
View Code
example.frm Form1
Properties - Form1
My Briefcase
Start
example.m...
Microsoft V...
Properties
Form1
Visual Basi...
3:22 PM

Form1 Form

Auto3D	False
AutoRedraw	False
BackColor	&H80000005&
BorderStyle	2 - Sizable
Caption	Form1
ClipControls	True
ControlBox	True
DrawMode	13 - Copy Pen
DrawStyle	0 - Solid
DrawWidth	1
Enabled	True

图10.13　增加 Rename 按钮

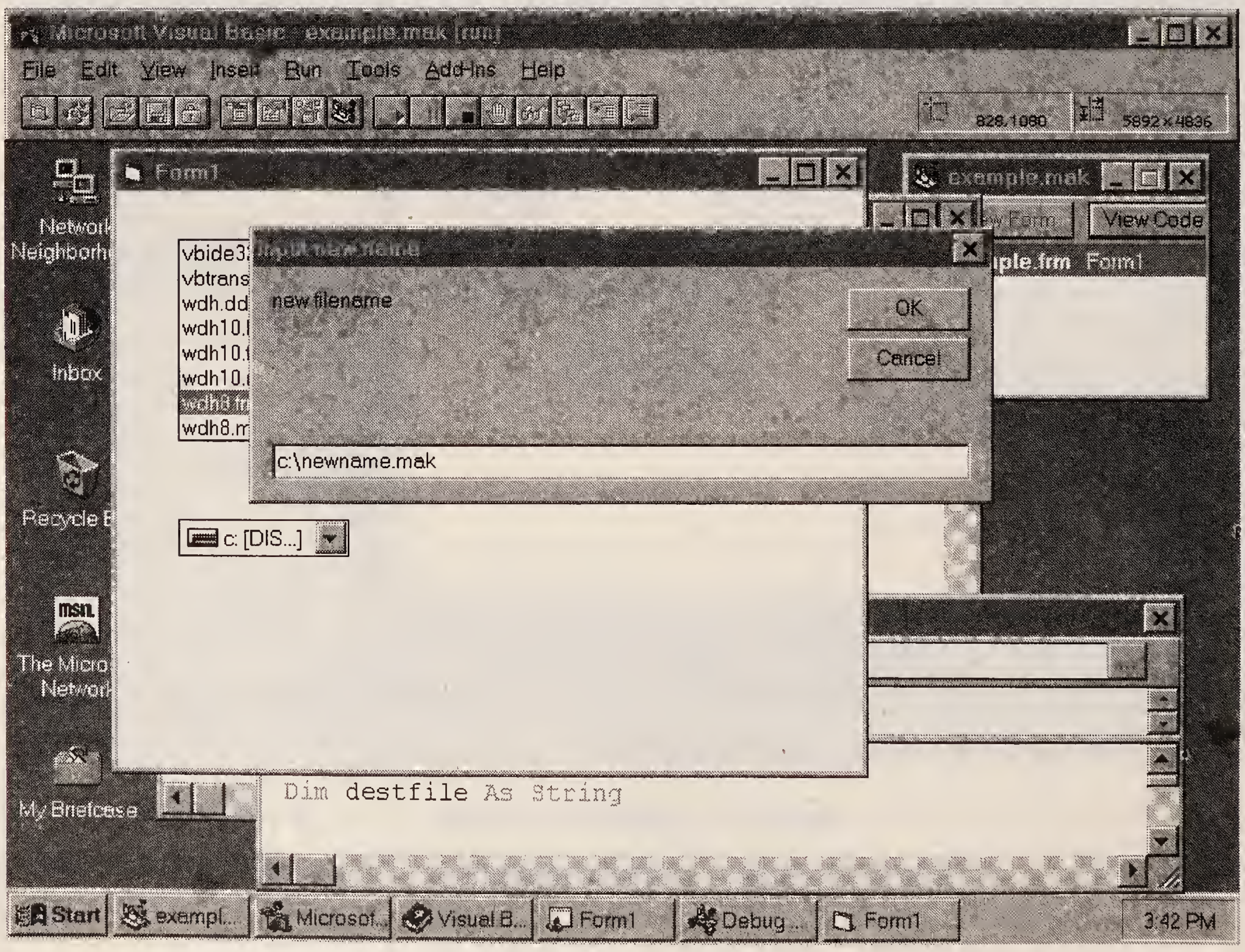
用户双击 Rename 按钮打开代码窗口并在 CmRename _ click()事件过程中编写以下代码：

```
Sub CmRename _ click()
    If File1.FileName=""Then
```

```
        MsgBox "Choose a file"
    End If
    If Right(File1.Path,1)<>"\"Then
        FileSource=File1.Path & "\" & File1.FileName
    Else
        FileSource=File1.Path & File1.FileName
    End If
    On Error GoTo ErrHandler
    RetVal=InputBox("New filename","input new name")
    Name FileSource As RetVal
    File1.Refresh
Exit Sub
    ErrHandler:
    If Err=55 Then
        MsgBox "Cann't rename a open file"
    Else
        MsgBox "Sepcify a file"
    End If
End Sub
```

在上面代码的第十行用户可以看到有一个错误处理语句:“On Error GoTo ErrHandler”。该语句起到一个错误陷阱的功能,当程序运行时发生错误不致于使系统崩溃,而是接着执行以 ErrHandler 为标记的语句产生有关错误信息。Err=55表示文件已被打开。

按 F5键运行程序,选择要更名的文件,然后单击 Rename 命令按钮,这时出现一个输入框,见图10.14。


new filename

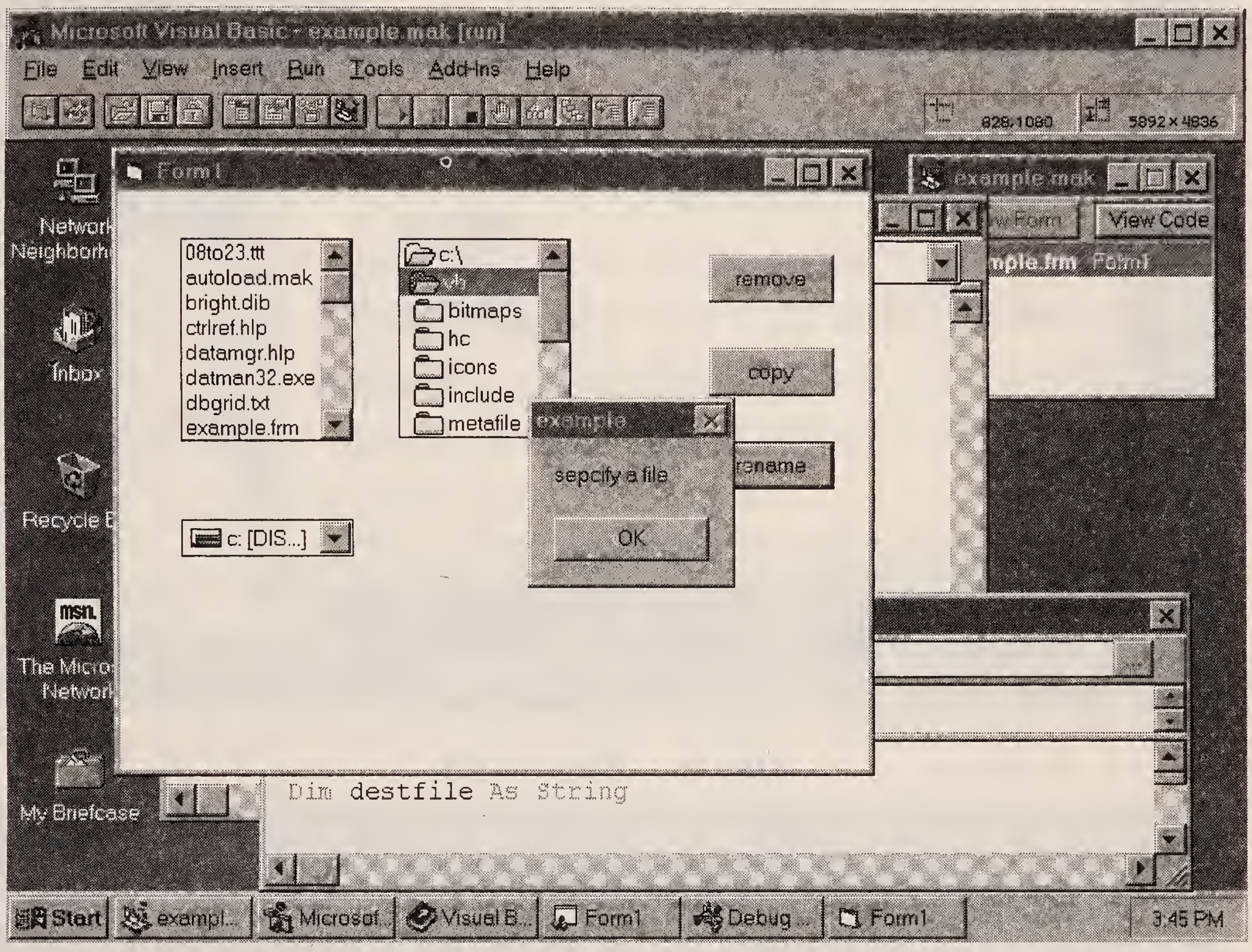
c:\newname.mak

```
Dim destfile As String
```

Microsoft Visual Basic - example.mak [run]
File
Edit
View
Insert
Run
Tools
Add-Ins
Help
828,1080
5892 x 4836
Form1
example.mak
View Code
ple.frm Form1
input new name
OK
Cancel
vbide3
vbtrans
wdh.dd
wdh10.
wdh10.
wdh10.
wdh8.fr
wdh8.m
c: [DIS...]
Network Neighborh
Inbox
Recycle B
The Micro Network
My Briefcase
Start
exampl...
Microsof...
Visual B...
Form1
Debug ...
Form1
3:42 PM

图10.14 输入框

用户可以在输入框中键入新文件名及存放路径，例如 C:\WIN95\NEWNAME.MAK，然后单击 OK 即可实现文件的更名。如果用户开始就没有选择一个文件而在输入框键入新文件名和路径，然后单击 OK 按钮，这时会出现一个消息框提示用户指定一个文件名，见图10.15。


Microsoft Visual Basic - example.mak [run]
File Edit View Insert Run Tools Add-Ins Help
828, 1080
5892 x 4836
Form1
08to23.ttt
autoload.mak
bright.dib
ctrlref.hlp
datamgr.hlp
datman32.exe
dbgrid.txt
example.frm
c:\
vb
bitmaps
hc
icons
include
metafile
c: [DIS...]
remove
copy
rename
example
OK
example.mak
View Form
View Code
example.frm Form1
Network Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Start
exampl...
Microsof...
Visual B...
Form1
Debug ...
Form1
3:45 PM

sepcify a file

```
Dim destfile As String
```

图10.15　确定文件的消息框

10.3.5　运行程序

在第八章我们已讲到用 Shell 函数可以运行一个以.BAT、.COM 或.EXE 为扩展名的应用程序。如果想使 EXAMPLE.MAK 工程文件也有运行一个应用程序的功能，打开 EXAMPLE.MAK 工程文件并在窗体上增加一个命令按钮，设置它的 Name 属性和 Caption 属性分别为 Cmrun 和 Run，见图10.16。

双击 Run 命令按钮打开代码窗口，在 cmrun _ click()事件过程编写以下代码以实现运行应用程序的功能。

```
Sub CmRun _ click()
    If Right(File1.Path,1) <>"\"Then
        FileRun=File1.Path & "\" & File1.FileName
    Else
        FileRun=File1.Path & File1.FileName
    End If
        RetVal=Shell(FileRun,1)
End Sub
```

按 F5运行程序，选择 C:\WIN95\CALC.EXE 文件，然后单击 Run 命令按钮，就可以运行计算器程序，见图10.17。

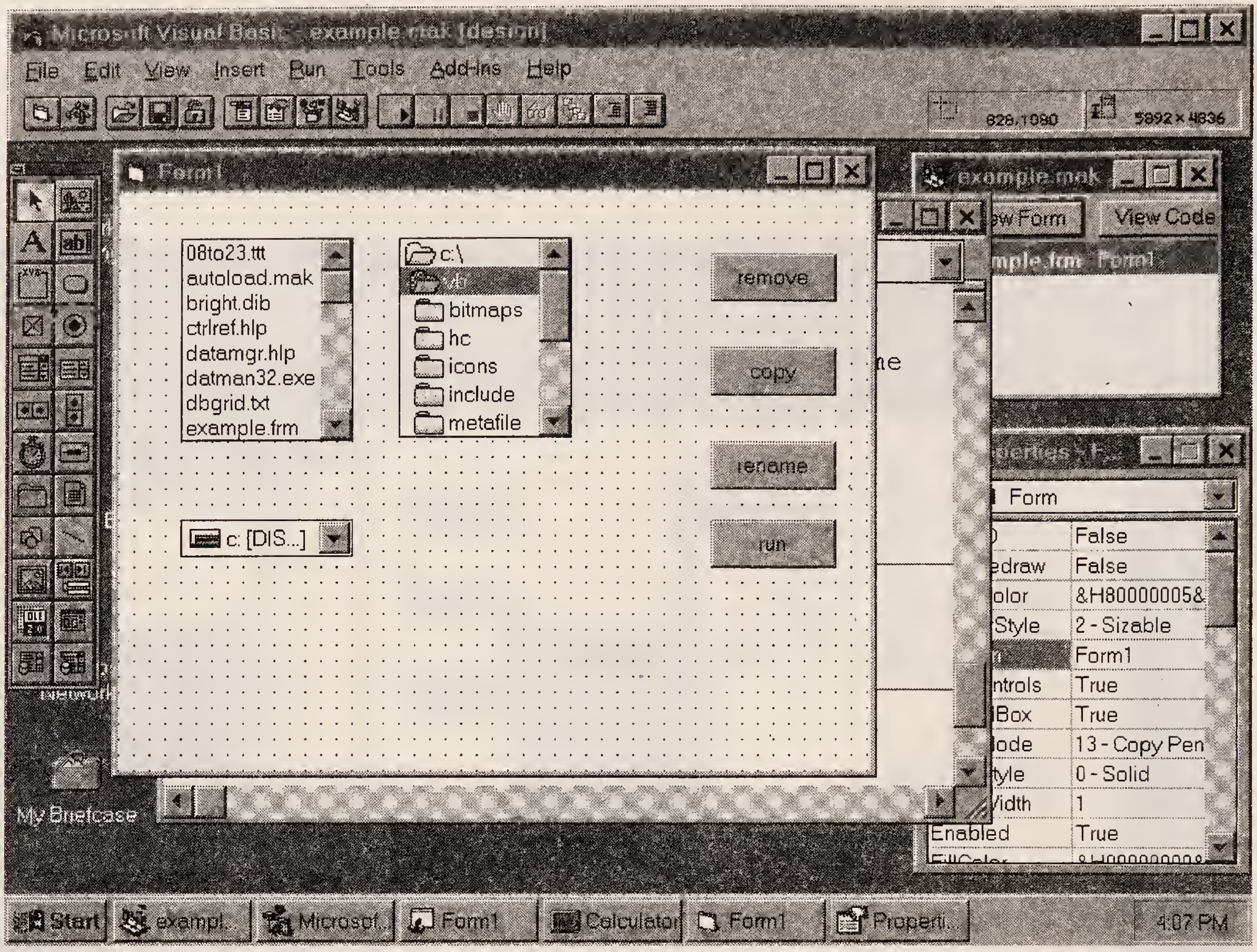
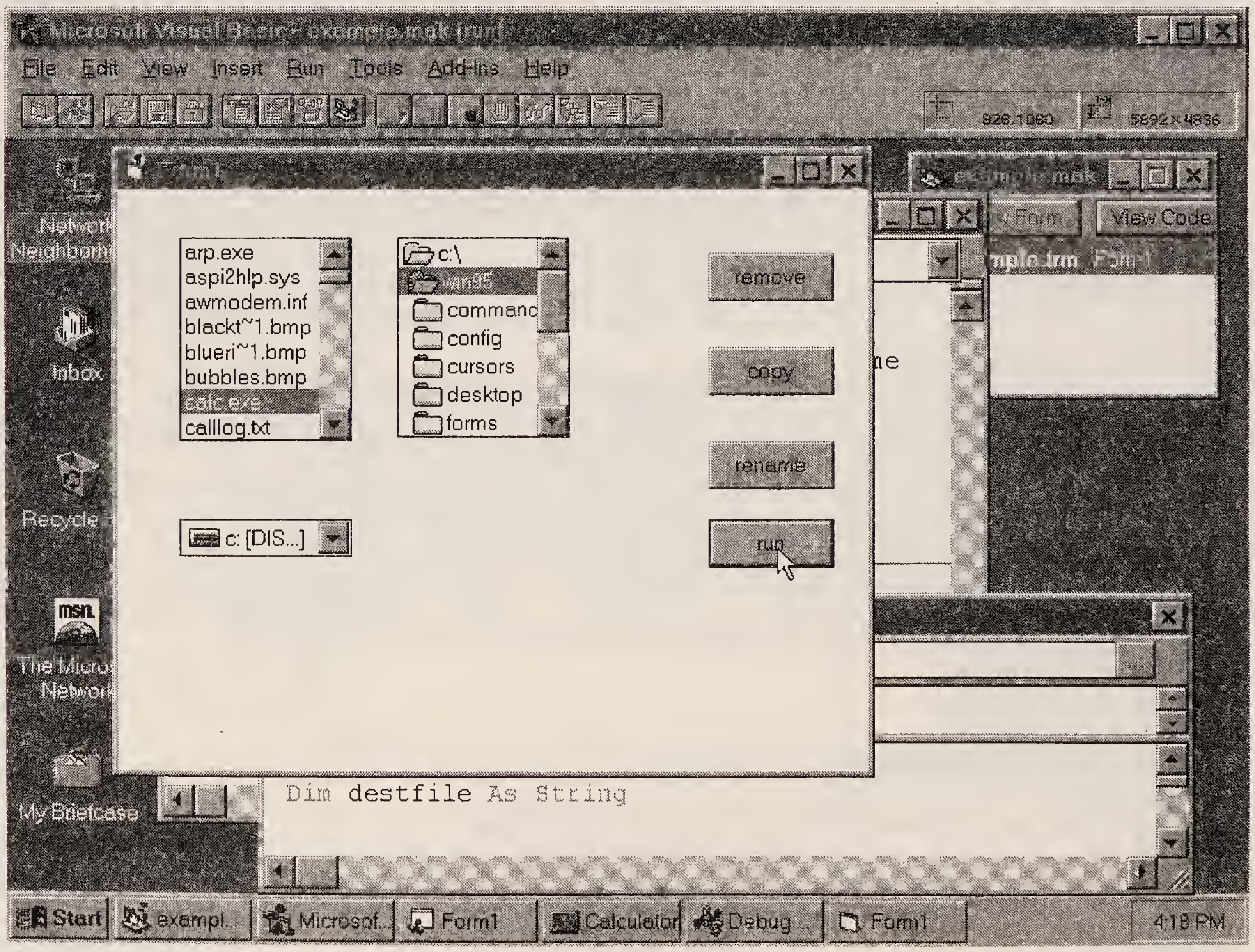

Microsoft Visual Basic - example.mak [design]
File Edit View Insert Run Tools Add-Ins Help
828,1080
5892 x 4836
Form1
08to23.ttt
autoload.mak
bright.dib
ctrlref.hlp
datamgr.hlp
datman32.exe
dbgrid.txt
example.frm
c:\
vb
bitmaps
hc
icons
include
metafile
c: [DIS...]
remove
copy
rename
run
example.mak
View Form
View Code
Properties - F...
Form
My Briefcase
Start
exampl...
Microsof...
Form1
Calculator
Form1
Properti...
4:07 PM

图10.16　增加 Run 按钮


Microsoft Visual Basic - example.mak [run]
File Edit View Insert Run Tools Add-Ins Help
828,1080
5892 x 4836
Form1
arp.exe
aspi2hlp.sys
awmodem.inf
blackt~1.bmp
blueri~1.bmp
bubbles.bmp
calc.exe
calllog.txt
c:\
win95
command
config
cursors
desktop
forms
c: [DIS...]
remove
copy
rename
run
example.mak
View Form
View Code
Network Neighborhood
Inbox
Recycle
The Microsoft Network
My Briefcase
Start
exampl...
Microsof...
Form1
Calculator
Debug...
Form1
4:18 PM

```
Dim destfile As String
```

图10.17　运行计算器程序

现在 EXAMPLE. MAK 工程文件已具有了删除、拷贝、更名和运行功能，或许用户还想添加一个 Cancel 命令按钮用来在做了某些操作之后结束程序的运行，要想完成此项工作很简单，只需添加一个命令按钮到窗体上并编写以下代码即可：

```
Sub CmCancel _ Click()
    End
End Sub
```

用户再把EXAMPLE. MAK文件存盘就拥有了一个功能较为完善的工程文件，如果再用File菜单的Make EXE File命令就可以生成一个可执行的应用程序EXAMPLE. EXE。

10.4 顺序文件

如果用户是处理文本文件，顺序存取方式或许是读取和存储文件中信息的最好选择，可以编写很简单的过程来读写文本的内容。然而顺序存取也有其缺点，如果文件存储的大部分是较长的数据，顺序存取会浪费存储空间。如果不是从开头，而是从文件的任意部位存取文件，顺序文件存取同样不是最好的选择。如果用户再经常修改文件内容，同样也不适于采取顺序文件存储方式，用户应根据具体情况而有选择地使用。

10.4.1 打开和关闭顺序文件

用户要想对顺序文件进行有关操作，必须要学会怎样打开顺序文件。Visual Basic提供了一个Open语句，使用该语句时用户可以以顺序、随机或二进制方式打开文件。在对文件执行任何I/O操作之前，必须打开此文件，Open为I/O分配一个文件缓冲区，并确定使用缓冲区的访问模式。Open语句的语法格式为：

```
Open file[Formode] [Access access] [lock] As [#] filenumber [len=reclen]
```

其中file代表文件名或路径，mode指定文件模式的保留字：Append、Binary、Input、Output或Random。access指定在的打开文件上允许哪种操作：Read(只读)、Write(只写)还是ReadWrite(读写)。filenumber代表一个在1到511之间的整数，用以作为标识符，在文件打开时标识该文件，用户可以把这项参数等同于在其他I/O操作中标识文件时用的“句柄”。reclen对于以随机访问方式打开的文件，指定记录长度；对于顺序文件，reclen指定缓冲区的字符数。lock指定了其他过程在打开文件上允许哪种操作。

如果文件不存在，当文件以Append、Binary、Output或Random模式打开文件时，将会创建一个文件。如果省略参数mode，缺省模式为Random。在以顺序文件存取时，可以在Open语句中使用以下三种模式：

Input—该操作使得可以从文件中读取
Output—该操作使得可以向文件中写入
Append—该操作使得可以向文件结尾追加数据

如果想打开一个文件进行读取，则使用Input模式，下面是一个例子：

```
Open "project. mak" for input As #1
```

本例打开名为project. mak的工程文件作为输入并且指定数字1作为其句柄名。在打开文件之后，便可以从文件中读取字符或一行行文本。

用户在打开一文件做了某些处理之后还要关闭文件，Visual Basic提供的Close语句用来关闭一文件。Close语句的语法格式为：

```
Close [[#]filenumber][,[#]filenumber]...
```

其中 filenumber 参数是 open 语句中用来打开文件所用的标识符，即前面所讲的句柄，可以使用任何数字表达式，只要它等于打开的文件号，当 Close 语句没有参数时，关闭所有的已被打开的文件。当执行 Close 语句时，与 filenumber 的相对应的文件被关闭，可以用相同或不同的文件号再重新打开此文件，例如：

```
Open "C:\CONFIG.SYS" For Output As #1
Close #1
Open "C:\CONFIG.SYS" For Output As #2
```

如果该文件不存在时企图打开文件读入，Visual Basic 将产生一个错误。如果打开一个用于写出的文件，Visual Basic 总是创建一个新文件，用同样的名字覆盖现存文件。如果用户在打开文件写之前希望检查现存文件，可以使用 Dir $ 函数，用一个文件名作参数调用 Dir $ 函数，函数将返回文件名的拷贝（表示该文件存在）或返回一个空字符串（表明没有该现存文件），下段程序表明了 Dir $ 函数的使用。

```
If Dir $ ("WW.TXT") <> " " Then
    If InputBox("WW.TXT exists,Delete it?") <> "Yes" Then
            Exit Sub
    End If
End If
Open "WW.TXT" For Output As #3
```

10.4.2 读和写顺序文件

1. 读顺序文件

为了从文件中读取信息，必须使用对应文件名以 input 模式打开文件，如果文件不在当前活动目录下，还必须用全路径名。同时，还要给出在程序中使用的文件标识符（句柄），取得有效标识符的最简便的方法是使用 FreeFile 函数，该函数返回一个能被 Open 语句使用的文件标识符。因此，仅用如下两行语句即可。

```
FileNumber%=FreeFile
Open FileName For Input As # FileNumber%
```

Visual Basic 提供 Input 和 Line Input 两个语句用来读取已被打开的文件的有关信息。Input 语句用来从已被打开的文件中读取数据并把数据安排给变量，它的语法格式为：

```
Input # filenumber,varlist
```

其中 filenumber 为一有效的文件标识符，varlist 是从文件读取的分配数值的变量表（用逗号隔开）。变量表中的变量不能是数值变量，但可以是描述数组元素的一个变量，也不能是一个对象变量。文件的数据必须同变量表的排列顺序一样，文件中的数据与变量的类型必须相匹配。当 varlist 中的一个变量为一个 String 时，如果遇到的第一个非空白字符，则认为是字符串的开始，当遇到逗号、空白或行尾时，即认为字符串结束（有双引号限制的字符串除外）。下面的代码就是 Input 语句从文件中读取非数字值输给一个变量：

```
Sub Form_Click()
    Dim FileData As String
    Dim Msg As String
Dim NL As String                        ' Declare variables
  NL=Chr(10)
MakeDataFile
Open "TESTFILE" For Input As #1
Do while Not EOF(1)
      Input #1,FileData
      Msg=Msg & FileData & NL
Loop
Close #1
MsgBox Msg
Kill "TESTFILE"
End Sub
Sub MakeDataFile
     Open "TESTFILE" For Output As #1
     Write #1,"This is test line1."
     Write #1,"This is test line2."
     Close #1
 End Sub
```

注意　函数 EOF(表示文件结束)说明是否已读完了数据，它只带有一个参数(文件标识符)，并返回一个布尔值，返回 True 说明读完所有数据。

Input＃语句只能读出第一个逗号或空格之前的信息，有时用户需要读出一行信息，这时可使用 Line Input＃语句，因为它忽略任何空格和逗号，能够接收回车键之前的所有输入信息。Line Input＃语句的语法格式为：

Line Input＃ filenumber，varname

其中 filenumber 是文件标识符(句柄)，varname 为变量名，用于从文件中接收一行文本。下面的代码是用 Line Input＃语句读取并显示 CONFIG. SYS 文件的第一行。

```
Sub Form _ Click()
Dim Drive As String,FileNumber As String
Dim Msg As String,NL As String,Textline As String
Msg="Which drive does your operating system boot from"
Drive=Ucase(Left(InputBox(Msg),1))
If Drive="C" or Drive="A" Then
     Filename=Drive & ":\CONFIG. SYS"
     If Dir(FileName) <> "" Then
       Open FileName For Input As #1
Line Input #1,Textline
Close #1
     Msg="The first line of your CONFIG. SYS file is:"
     Msg=msg & NL && Textline & "."
Else
     Msg="Could not find a CONFIG. SYS file on drive"
     Msg=Msg & Drive & ":"
```

```
    End If
    Else
        Msg="You didn't provide valid boot drive"
        Msg=Msg & "normally boot from A: or C:drive"
    End If
    MsgBox Msg
    End Sub
```

除了能够用 Input # 和 Line Input # 语句读文件以外，Visual Basic 还提供了一个 Input $ 函数用来从顺序文件中读取返回的字符串，该函数的语法格式为：

```
Input $ (n, # filenumber)
```

n 参数表示从文件中读取的字符数，它必须小于65535，filenumber 参数表示用 Open 语句打开文件时的文件标识符。

同 Input # 语句不一样，Input $ 返回读取的所有字符，包括逗号、回车、换行、引号和前导空格。Input $ 函数只有在以 Input 或 Binary 打开文件时使用。下面的例子是使用 Input 的函数从文件读取字符串的情况：

```
例1 Open "WWW.TXT" For Input As #1
    Input#(512,#1)
    Close #1
例2 Open "WW.TXT" For Input As #1
    Text1.Text=Input(LOF(1),#1)
    Close #1
```

例1是从文件 WWW.TXT 读取512个字节；例2中用 LOF 函数获取文件长度的字节数，其中两个数字1是指文件的句柄，这样 Input $ 函数便从文件中读取整个文件并把它赋给文本框的 Text 属性。

2. 写顺序文件

写顺序文件和读顺序文件一样容易。Print # 语句用于写顺序文件，其方法很像用 Print 方法在窗体上显示正文，它的语法格式如下：

```
Print# filenumber,[outputlist] [{;1,}]
```

filenumber 参数表示 Open 语句打开文件时的句柄。expressionList 参数表示写到文件的数值或串表达式。“;1,”参数决定下一个字符开始打印的位置，分号代表下一字符紧随最后一个字符打印，逗号代表下一个字符在下一个打印区开始进行打印。如果省略 expressionList 参数，则 Print # 语句在文件中打印一空行，但逗号必须包括在内。

下面一例是用 Print # 语句向一个测试文件写入日期，注意分号与逗号的使用：

```
Sub Form _ Click()
  Dim FileData As String,Msg as String
  Dim NL as String                     'Declare variables
  NL=Chr(10)                           'Define newline
  Open "TestFile" For Output As #1     'Open file to write
```

```
     Print #1,″This is a test of print# statement.″
     Print #1,                                    ’print blank line to file
     Print #1,″Zone1″,″Zone2″                     ’print in two print zones″
     Print #1,″with no space between″ ; ″.″       ’Print two
                                                  ’Strings together
  Close #1
  Open ″TestFile″ For Input As #1                 ’Open file to read
  Do while Not EOF(1)
       Line Input #1,FileData                     ’Read a line of data
       Msg=Msg & FileData & NL
    Loop
    Close #1
    MsgBox Msg                                    ’Display message
    Kill ″TestFile″                               ’Remove file from Disk
  End Sub
```

除了能够用 Print 语句将数据写入顺序文件以外，Visual Basic 还提供了 Write# 将数据写入顺序文件，与 Print# 语句不同的是，Write# 语句在数据项和引号括起来的字符串之间插入了逗号，用户不必在列表中显式地写分隔符。Write# 语句的语法格式为：

```
Write# filenumber [,expressionlist]
```

其中由 filenumber 标识的文件必须按照 Output 或 Append 模式打开。Write# 语句在向文件写入 expressionlist 的最后一个字母时将插入一个换行符。下面一例是用 Write 语句写入两个变量到顺序文件中去：

```
Sub Form _ Click
   Dim Msg As String,NL As String
   Dim UseAge As String,UseName AS String         ’Declare variable
   NL=Chr(10)                                     ’Define new line
   Open ″TestFile″ For Output As #1               ’open fite for output
   Msg=″Enter your name″
   UseName=InputBox(Msg)                          ’Get user name
   Msg=″Enter your age″
   UseAge=InputBox(Msg)                           ’Get user name
   Write #1,UseName,UseAge                        ’Write data to file
   Close #1                                       ’Close file
                                                  ’Read the sample data file
   Open ″TestFile″ For Input As #1                ’open file to read
   Input #1,UseName,UseAge                        ’read data
   Msg=″The name″ & UseName & ″ was read from file″
   Msg=Msg & ″,″ & UseAge & ″is the age you give″
```

```
        MsgBox Msg                                    'Display message
        Kill "TestFile"                               'remove file
      End Sub
```

如果只送单个信息块到文件，那么就可以交替使用 Print＃和 Write＃语句，例如：

```
Print #1,"Hello"
Write #1,"Hello"
```

它们都实现了送入单个信息块到文件中，对于这两种情况也都能用 Input＃命令读取信息。仅当同时送入多块信息时才出现文件的明显差异。例如，要送入三个数值到文件中，使用：

```
Print #1,1,2,3
```

将同时产生大量多余的空格，而命令：

```
Write #1,1,2,3
```

将数值和适当的逗号送入文件，既节省了存储空间，又容易读取，它等效于一条复杂的语句：

```
Print #1,1; "," ;2; "," ;3
```

简而言之，Write＃语句和 Input 语句一起配对使用，而 Print＃和 LineInput＃一起配对使用。

3. 向顺序文件中添加信息

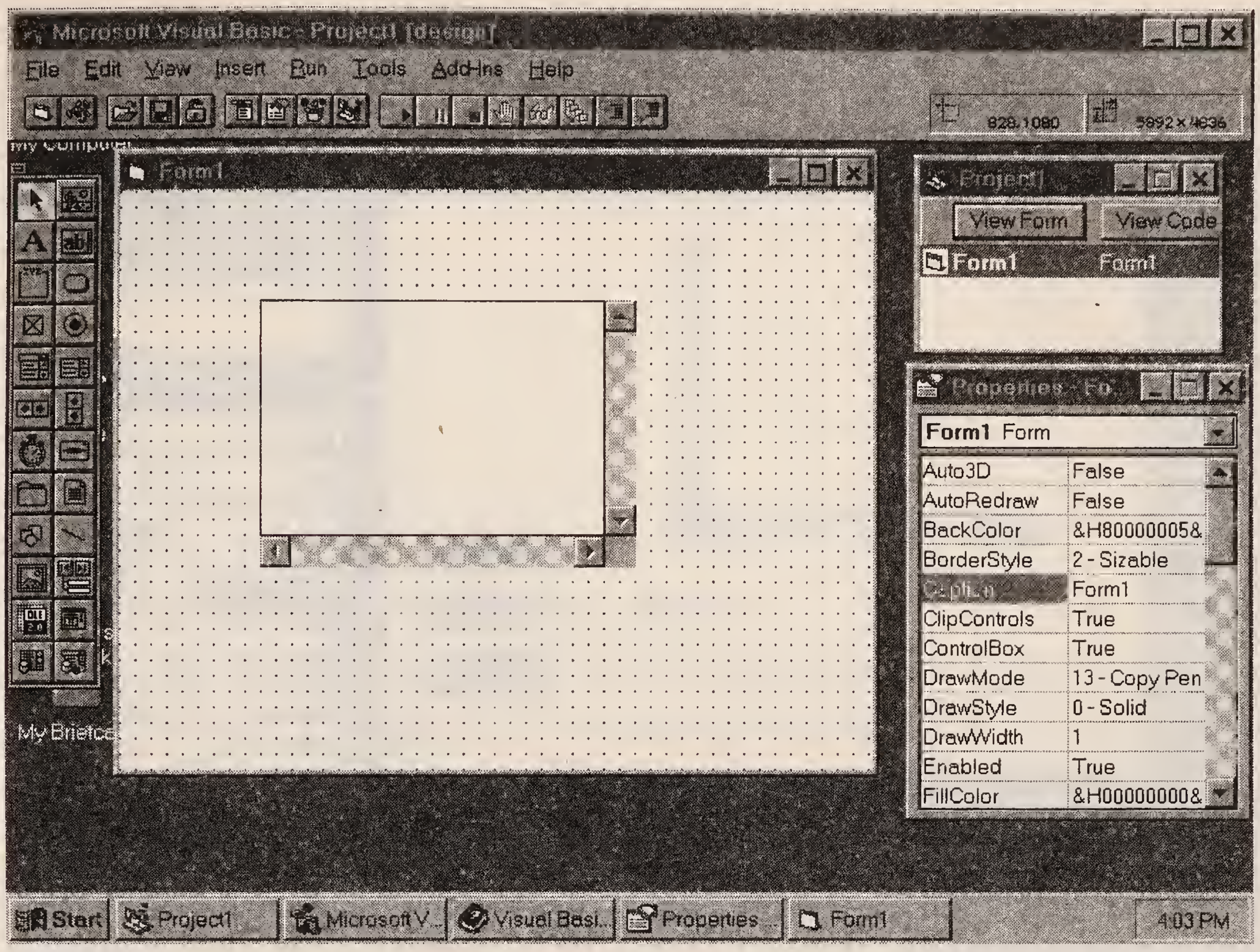
前面已经提到，使用 Append 模式可以向顺序文件追加信息。当用户用 Append 模式向顺序文件追加信息时，Visual Basic 打开文件（如果不存在，就建立一个）并设置适当的缓冲区，然后把文件指针设置到文件末端，最后用一个 Print＃语句或 Write＃语句追加信息到文件尾部，例如：

```
Open "TestFile" For Append As #1
   Msg$ ="Add information to a file"
Print #1,Msg$
Close #1
```

10.4.3 一个编辑 Autoexec.bat 的应用程序

或许用户有时要更改 Autoexec.bat 文件，下面我们将用上节讲得有关知识编写一个编辑 Autoexec.bat 文件的应用程序。

为了显示 Autoexec.bat 文件的内容，文本框的 Multiline 属性应该设置为 True，这样文本框就可以显示多行信息。为了能够方便用户查看文本框的内容，应该把文本框的 ScrollBars 属性设置为3-Both，这时在文本框上出现水平滚动条和垂直滚动条，见图10.18，用户单击滚动条就可以滚动文本框内文本。


Microsoft Visual Basic - Project1 [design]
File
Edit
View
Insert
Run
Tools
Add-Ins
Help
828,1080
5992 x 4636
My Computer
Form1
Project1
View Form
View Code
Form1
Form1
Properties - Fo
My Briefcase
Start
Project1
Microsoft V...
Visual Basi...
Properties ...
Form1
4:03 PM

Form1 Form

Auto3D	False
AutoRedraw	False
BackColor	&H80000005&
BorderStyle	2 - Sizable
Caption	Form1
ClipControls	True
ControlBox	True
DrawMode	13 - Copy Pen
DrawStyle	0 - Solid
DrawWidth	1
Enabled	True
FillColor	&H00000000&

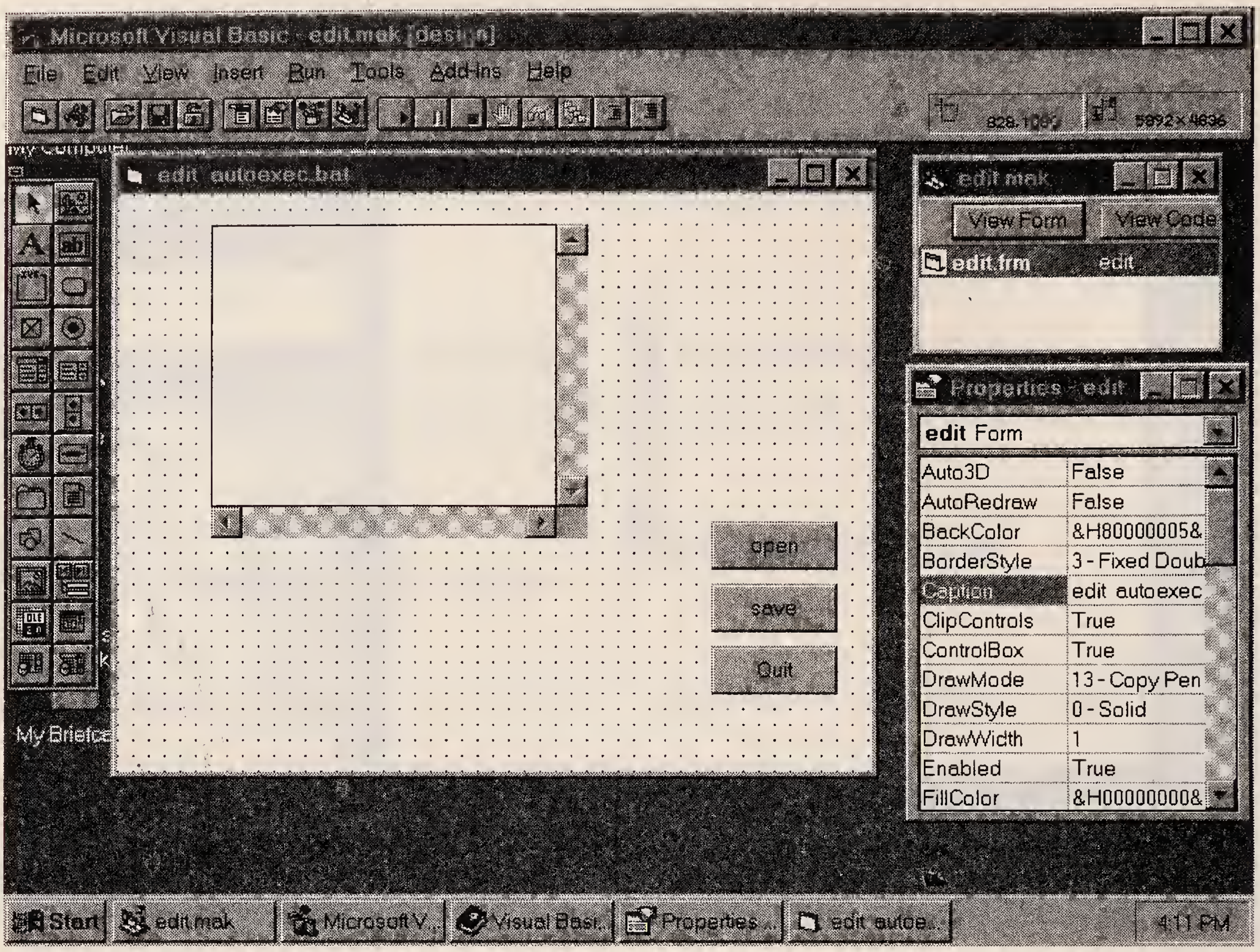
图10.18 带有滚动条的文本框

再向窗体增加三个命令按钮，设置这些控件的属性如下表，建立如图10.19的初始化界面。

控件	属性	设置值
窗体	Borderstyle	3-Fixed Double
	Caption	Edit Autoexec. bat
	Name	Edit
文本框	Multiline	True
	Name	TxtEdit
	ScrollBars	3-Both
	Text	空白
命令按钮1	Caption	Open
	Name	CmOpen
命令按钮2	Caption	Save
	Name	CmSave
命令按钮3	Caption	Quit
	Name	CmQuit

现在用户如果单击图10.19中的 Open 按钮时，文本将不会发生任何变化，因为还没有编写相应的代码。双击窗体的空白处打开代码窗口，选择对象列表框的 General 对象，在声明部分加入以下全程声明变量：

```
Dim TextChanged As Integer
Dim AutoFound As Integer
Dim Enter As String
```


Microsoft Visual Basic - edit.mak [design]
File Edit View Insert Run Tools Add-Ins Help
828, 1065
5892 x 4635
My Computer
edit autoexec.bat
open
save
Quit
My Briefcase
edit.mak
View Form
View Code
edit.frm
edit
Properties - edit
Start
edit.mak
Microsoft V...
Visual Basi...
Properties ...
edit autoe...
4:11 PM

edit Form

Auto3D	False
AutoRedraw	False
BackColor	&H80000005&
BorderStyle	3 - Fixed Doub
Caption	edit autoexec
ClipControls	True
ControlBox	True
DrawMode	13 - Copy Pen
DrawStyle	0 - Solid
DrawWidth	1
Enabled	True
FillColor	&H00000000&

图10.19　初始化界面

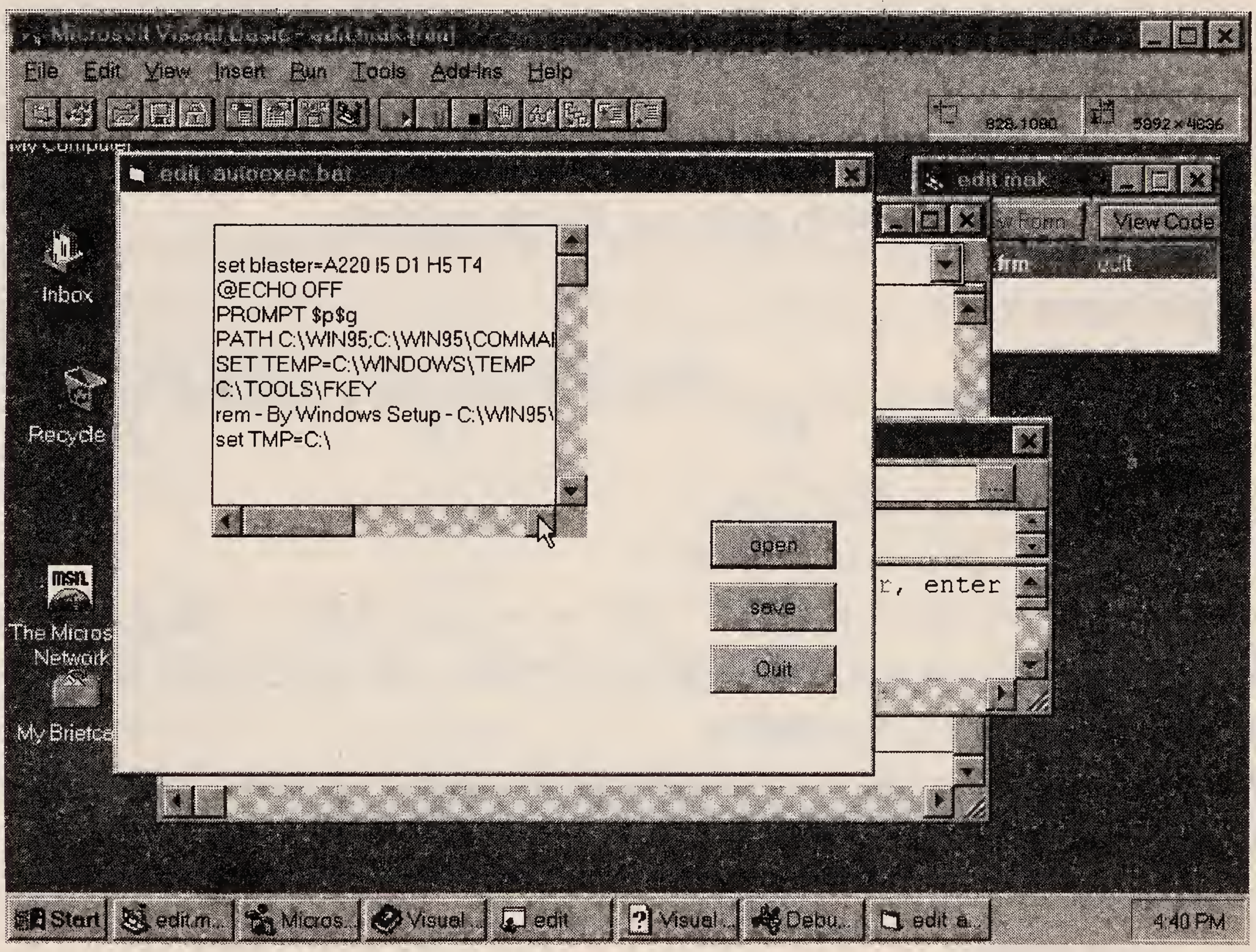
然后选取其他的对象和过程编写如下代码：

```
Sub Form _ Load()
  Textchanged=False
  AutoFound=False
  Enter=Ch $ (13)+Chr $ (10)
  End Sub
Sub CmOpen _ Click()
  If AutoFound Then
         Beep
     Exit Sub
  End If
Open "C:\AUTOEXEC.BAT" For Input As #1
Do while Not EOF(1)
      Line Input #1,Cline
        TxtEdit.Text=TxtEdit.Text+Cline+Enter
Loop
Close #1
AutoFound=True
End Sub
Sub CmSave _ Click()
  If Not(AutoFound And TextChanged)Then
       Beep
       Exit Sub
```

```
    End If
    Oepn "C:\AUTOEXEC.BAT" For Output As #1
    Print #1,TxtEdit.Text
    Close #1
End Sub
Sub CmQuit _ Click()
     End
End Sub
Sub TxtEdit _ Change()
    TextChanged=True
End Sub
```

代码中的 Beep 是为使计算机扬声器发生警报声。

当用户按 F5键运行程序后单击 Open 按钮，这时文本框显示出 Autoexec. bat 文件的全部内容，见图10. 20。如果文本框显示不下 Autoexec. bat 的全部文本，可以单击滚动条浏览。


Microsoft Visual Basic - edit.mak [run]
File
Edit
View
Insert
Run
Tools
Add-Ins
Help
828,1080
5892 x 4836
My Computer
edit autoexec.bat
edit.mak
View Form
View Code
frm
edit
Inbox
Recycle
The Microsoft Network
My Briefcase
open
save
Quit
r, enter
Start
edit.m...
Micros...
Visual...
edit
Visual...
Debu...
edit a...
4:40 PM

```
set blaster=A220 I5 D1 H5 T4
@ECHO OFF
PROMPT $p$g
PATH C:\WIN95;C:\WIN95\COMMAI
SET TEMP=C:\WINDOWS\TEMP
C:\TOOLS\FKEY
rem - By Windows Setup - C:\WIN95\
set TMP=C:\
```

图10. 20　显示 Autoexec. bat 文本内容

用户可以通过单击文本框把焦点移动文本框中，这时就可以编辑 Autoexec. bat 文件，然后单击 Save 按钮保存编辑结果，单击 Quit 可以退出程序的运行。

10.5 随机文件

顺序文件对于文本文件来说是不错的，但是在读取其他类型的文件时便不是最好的选择。大多数用户在顺序文件输入时会碰到两个问题：其一是用正文表示数据的空间效率总是不高。例如，可以在4个字节中存储数值43. 2567，但在顺序文件中按正文存储该数值需7个字

节；其次是有时需要按非顺序方式处理信息。用户可用随机文件存储解决以上问题。

10.5.1　打开和关闭随机文件

随机文件的建立通常分为四个步骤：(1) 打开文件；(2) 定义数据类型；(3) 将数据写入磁盘中；(4) 关闭文件。

打开随机文件时，Open 语句中的模式必须为 Random，它的格式如下：

```
Open filename For Random As #filenumber Len=记录长度
```

随机文件在打开时必须指定记录的长度，并且随机文件只要打开一次就可以同时读和写。随机文件的记录长度是将记录中各域所占的字节数相加起来，用户最好设置最大的可能长度。为了使随机文件每个记录都有相同的长度，用一变量名代替整个记录中每个域，然后用 Type/End Type 语句定义记录的各个域的类型并指定每个域的字节数。例如下面就是用 Type/End Type 语句定义一个长度为36个字节的记录：

```
Typed StudentRecord
        StudentName As String * 12
        StudentNo As String * 12
        StudentSchool As String * 12
End Type
```

关闭随机文件与顺序文件相同，使用 Close 语句来关闭文件。

10.5.2　读和写随机文件

Visual Basic 提供了 Get # 和 Put # 语句用于读写随机文件，这两种语句的语法格式如下：

```
Get # filenumber,[recordnum],varname
Put # filenumber,[recordnum],varname
```

filenumber 是一个有效的文件标识符(句柄)，recordnum 是表示一个记录数；varname 表示一个读入或写入数据的有效变量名。在随机文件中第一个记录的位置是1而不是0，如果用户省略记录数，则从下一个记录开始读取。

下面一个例子是用 Get 语句从文件中读取数据到变量中，这个例子假设 TESTFILE 文件含有五个用户定义的记录。

```
Type Record
      ID As Integer
      Name As String * 20
End Type                                    'Define user-defined Type
Dim MyRecord As Record
'Open sample file for random access
Open "TESTFILE" For Random As #1 Len=Len(MyRecord)
'read the sample file using the Get statement
Position=3                                  'Define record number
Get #1,Position,MyRecord
Close #1
```

下面一个例子用 Put 语句把数据写到文件中，被写入文件的是用户定义的五个记录。

```
Type Record
    ID As Integer
    Name As String * 20
End Type
Dim MyRecord As Record
Open "TESTFILE" For Random As #1 Len=Len(MyRecord)
For RecordNumber=1 To 5                       'Loop 5 times
MyRecord. ID=RecordNumber                     'Define ID
MyRecord. Name="MyName" & RecordNumber        'Create a string
Put #1,RecordNumber,MyRecord
Next RecordNumber                             'Write record to file
Close #1                                      'Close file
```

10.5.3　一个学生档案数据库程序

顺序文件对于文本文件来说是不错的，但在读取以某种数据类型存储的记录时，随机存取对于读取含有多条记录的文件是很合适的。本节将介绍用随机存储建立一个学生档案数据库管理系统。

首先，用户必须向窗体中添加组合框、文本框、标签和命令按钮等控件，组合框用来显示学生的姓名的列表，文本框用来显示有关学生的信息，例如姓名、年龄和性别等。设置各个控件的属性如下表，建立如图10.21所示的初始化界面。

对象	属性	设置
Form1	BorderStyle	3-Fixed Double
	Caption	Student database system
	Name	Student
Command1	Name	CmAdd
	Caption	Add
Command3	Name	CmQuit
	Caption	Quit
Label1	Alignment	1-Right Jusify
	Caption	Name
Combol	Name	CboName
	Style	2-Dropdown list
Label2	Alignment	1-Right Jusify
	Caption	Sex
Text1	Name	TxtSex
	Text	空白
Label3	Alignment	1-Right Jusify
	Caption	Weight
Text2	Name	TxtWeight
	Text	空白

（续表）

对象	属性	设置
Label4	Alignment	1-Right Jusify
	Caption	Height
Text3	Name	TxtHeight
	Text	空白
Label5	Alignment	1-Right Jusify
	Caption	Age
Text4	Name	TxtAge
	Text	空白
Label6	Caption	Student Database System
Command2	Name	CmDelete
	Caption	Delete

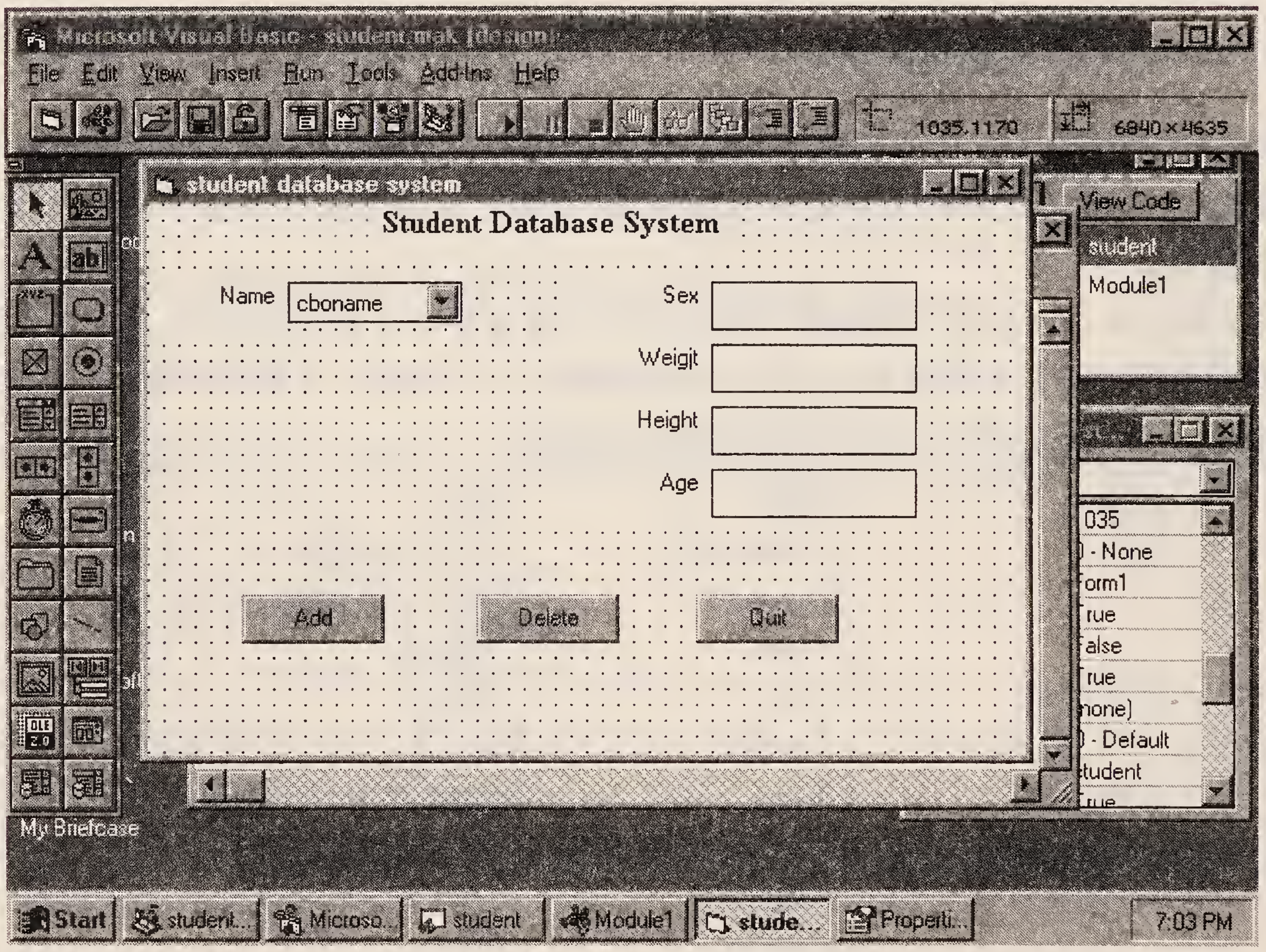

Microsoft Visual Basic - student.mak [design]
File Edit View Insert Run Tools Add-Ins Help
1035,1170
6840 x 4635
student database system
Student Database System
Name
cboname
Sex
Weigit
Height
Age
Add
Delete
Quit
View Code
student
Module1
My Briefcase
Start
7:03 PM

图10. 21　初始化界面

应用程序需要一些全程声明，因为全程变量和类型声明不能在窗体模块中进行，所以需要创建一个代码模块，从 File 菜单选取 New Module 命令。在代码窗口的全程声明段加入以下声明：

```
Type StudentType
    Name As String * 12
    Sex As Stirng * 12
    Weight As Integer
    Height As Integer
    Age as Integer
End Type
```

```
Global const Student _ Reclen=30
```

上面定义了一个常量使程序更易于阅读，看到常量名比看到30更能方便地理解该数值表示的意义。定义常量重要的意义是能够方便地修改程序。例如用户想增添一个域，只需改变该常量的值，而不必更改程序的每一部分。声明的其余部分局限于窗体模块内。关闭代码模块的代码窗口，在全程声明段加入以下三行：

```
Dim CurrentStudentLx As Integer
Dim StudentCount As Integer
Dim RecordModified As Interger
```

这些变量将分别用于声明当前学生记录的记录号、在文件中学生的总数以及用户是否修改了学生的记录。

在初始化系统时，程序应读文件中的每一个记录，并将所有学生名放在组合框的列表中。当用户选择一个名字时，程序应该取出该学生的整个记录并在文本框中显示出。如果用户修改了记录，修改后的记录也应该被重新保存到文件中。

```
Sub StudentInit()
Dim Student As StudentType
Open "C:\VB\STUDENT.DAT" for random As #1 Len=Student _ Reclen
StudentCount=LOF(1)/student _ RECLEN
For S=1 To StudentCount
Get #1,S,Student
CboName.Additem student.Name
Next S
End Sub
```

Open 语句用于打开或创建文件。函数 LOF 返回文件中字节的总数，用它除以记录长度得到总的记录数。然后用 Get 语句读取每一个记录，并将每个学生名填写到组合框的列表中。

当完成初始化时，用户可以选择学生名，显示学生的记录，然后删除这些记录，下面的代码完成这些功能：

```
Sub ShowStudent(lx As Integer)
  Dim Student As Studenttype
  Get #1,lx+1,Student
    CurrentStudentLx=lx
    CboName.ListIndex=lx
    TxtSex.Text=Student.Sex
    TxtWeight.Text=Str$(Student.Weight)
    TxtHeight.Text=Str$(Student.Height)
    TxtAge.Text=Str$(Student.Age)
        RecordModified=False
    End Sub

Sub UpdateCurrentStudent()
  Dim Student As StudentType
```

```
    If Not RecordModified Then
                Exit Sub
  End If
      Student. Name=CboName. Text
      Student. Sex=TxtSex. Text
      Student. Weight=Val(TxtWeight. Text)
      Student. Height=Val(TxtHeight. Text)
      Student. Age=Val(TxtAge. Text)
      Put #1,CurrentStudentLx+1,Student
             RecordModified=False
  End Sub
```

ShowStudent 过程用来显示一个学生的记录，该过程需要一个记录号参数。updatecurrentstudent()过程检查是否修改了记录。如果没有修改，则过程马上结束。如果用户已修改记录，每个域的当前值被读出，其结果将写回到磁盘文件中。

程序的其他任务包括创建新记录和删除旧记录，下面的代码将用于这两项功能的实现。NewStudent 函数用传递给它的名字和缺省值填写学生记录。它将 StudentCount 加1，说明增加了一个记录，并将该记录写回到磁盘文件中，在组合框列表中加入该学生的名字。由于新的记录总是写在学生的尾部，其列表框下标和以前的 StudentCount 一样，又当作函数值返回给调用过程。

```
  Function NewStudent(NewName As String) As Integer
  Dim Student As studenttype
      Student. Name=NewName
      Student. Sex="male"
      Student. Weight=0
      Student. Height=0
      Student. Age=0
      New Student=StudentCount
      StudentCount=StudentCount+1
      Put #1,StudentCount,Student
      CboName,Additem Student. Name
          RecordModified=False
      End Function
```

注意　到目前为止，所有的代码过程都在窗体的全程段中定义，其原因是每一过程都执行特定的任务，为了其他过程都可以调用这些过程，所以最好将它们定义为独立的单元。

```
  Sub CmDelete _ Click()
    Dim Student As StudentType ,S As integer
    'copy all but CurrentRecord to tempfile
    open "C:\VB\STUDENT. TMP" for Random As #2 Len=Student _ Reclen
    For S=1 To StudentCount
      If S<>CurrentStudentLx + 1 then
          Get #1,S,Student
          Put #2,S,Student
```

```
        End If
      Next S
      'Delete and rename
      Close #1
      Close #2
      Kill "C:\VB\STUDENT.DAT"
      Name "C:\VB\STUDENT.TMP" As C:\VB\STUDENT.DAT"
      cboName.Clear
        StudentInit
    End Sub
```

第一种重要关系是将程序启动和数据库初始化联系起来，所以要编写以下代码：

```
    Sub Form_Load()
      StudentInit
    End Sub
```

其次，还必须把用户从列表中选择学生名和显示相关的记录联系起来，并且必须连接Add按钮和创建新记录。以下两个过程完成以下两种连接：

```
    Sub CboName_Click()
      ShowStudent CboName.ListIndex
    End Sub
    Sub CmAdd_Click()
    Dim NewName As String
    NewName=InputBox("What's new name of student")
    If NewName="" Then
          Exit Sub
    End If
    ShowStudent Newstudent(NewName)
    End Sub
```

为了完善应用程序，用户必须编写代码以支持编辑学生记录。为此编写以下代码：

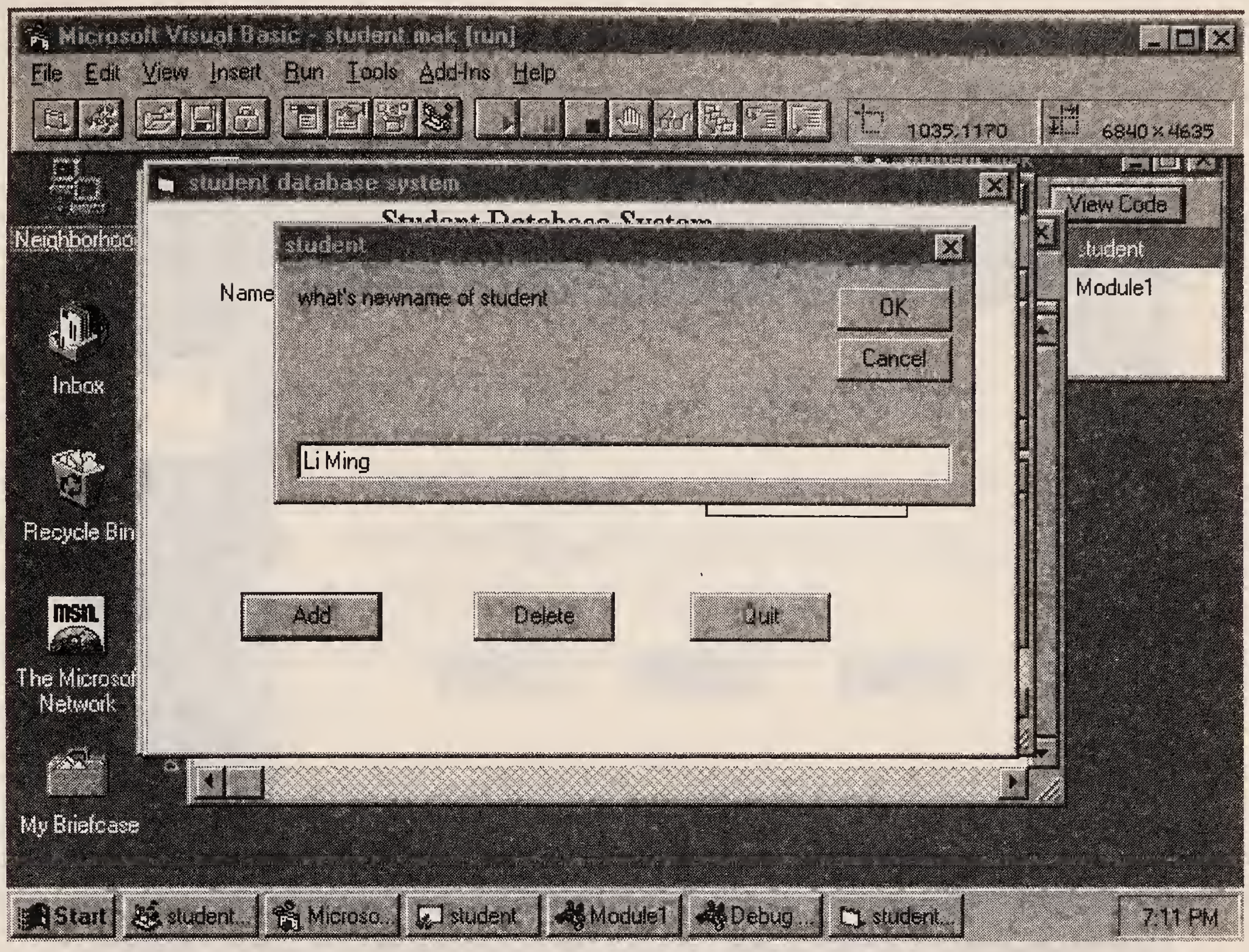
```
    Sub TxtSex_GotFocus()
      TxtSex.SelStart=0
      TxtSex.SelLength=Len(TxtSex.Text)
    End Sub
    Sub TxtWeight_GotFocus()
      TxtWeight.SelStart=0
      TxtWeight.SelLength=len(TxtWeight.Text)
    End Sub

    Sub TxtHeight_GotFocus()
      TxtHeight.SelStart=0
      TxtHeight.SelLength=Len(TxtHeight.Text)
    End Sub

    Sub TxtAge_GotFoucs()
      TxtAge.SelStart=0
    TxtAge.SelLength=Len(TxtAge.Text)
End Sub

Sub TxtSex_Change()
    RecordModified=True
End Sub

Sub TxtSex_LostFocus()
    UpdateCurrentStudent
End Sub

Sub TxtWeight_Change()
    RecordModified=True
End Sub

Sub TxtWeight_LostFocus()
    UpdateCurrentStudent
End Sub

Sub TxtHeight_Change()
    RecordModified=True
End Sub

Sub TxtHeight_LostFocus()
    UpdateCurrentStudent
End Sub

Sub TxtAge_Change()
        RecordModified=True
End Sub

Sub TxtAge_LostFocus()
        UpdateCurrentStudent
End Sub
```

为结束程序运行，还应该编写以下代码：

```
Sub CmQuit_Click()
        End
End Sub
```

到此为止，用户已做好了编程工作。现在可以检验一下程序的运程情况。

单击工具条上的 Start 按钮或按 F5键运行程序。由于 STUDENT.DAT 是新建的文件，该文件还没有包含学生的记录，所以组合框没有列出学生的姓名。用户可以单击 Add 按钮向文件中添加学生的记录。当用户单击 Add 按钮后，屏幕上弹出一个输入框，在框中可以键入学生的姓名(Li Ming)，见图10.22。

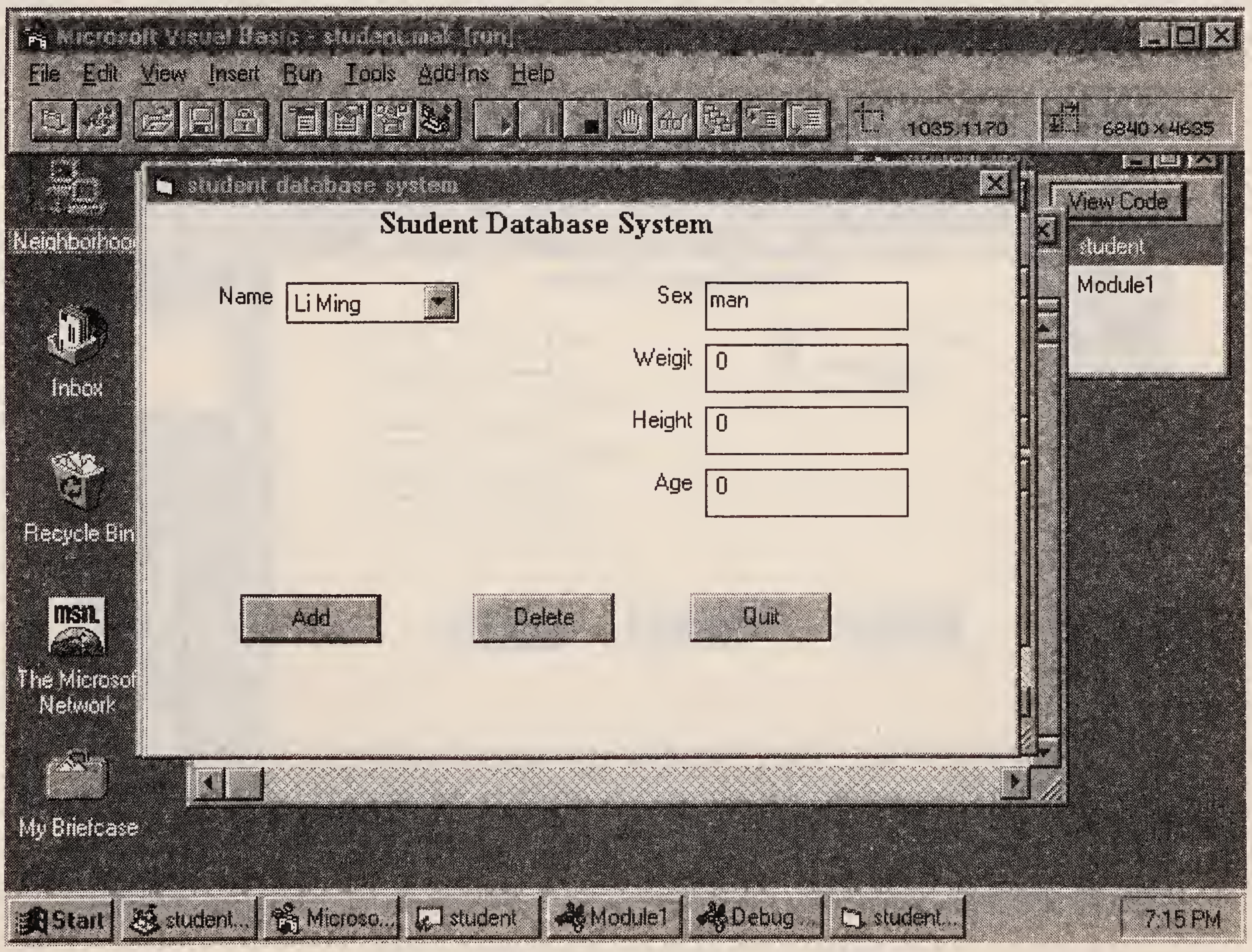

Microsoft Visual Basic - student.mak [run]
File Edit View Insert Run Tools Add-Ins Help
1035,1170
6840 x 4635
student database system
student
OK
Cancel
Add
Delete
Quit
View Code
student
Module1
Neighborhoo
Inbox
Recycle Bin
The Microsof Network
My Briefcase
Start
student...
Microso...
student
Module1
Debug...
student...
7:11 PM

what's newname of student

Li Ming

图10. 22　在输入框中键入新学生的姓名

单击输入框上的 OK 按钮，这时姓名添加到组合框中，同时文本框显示出该学生(Li Ming)有关项的缺省设置，见图10. 23。


Microsoft Visual Basic - student.mak [run]
File Edit View Insert Run Tools Add-Ins Help
1035,1170
6840 x 4635
student database system
Add
Delete
Quit
View Code
student
Module1
Neighborhoo
Inbox
Recycle Bin
The Microsof Network
My Briefcase
Start
student...
Microso...
student
Module1
Debug...
student...
7:15 PM

Student Database System

Name Li Ming

Sex man

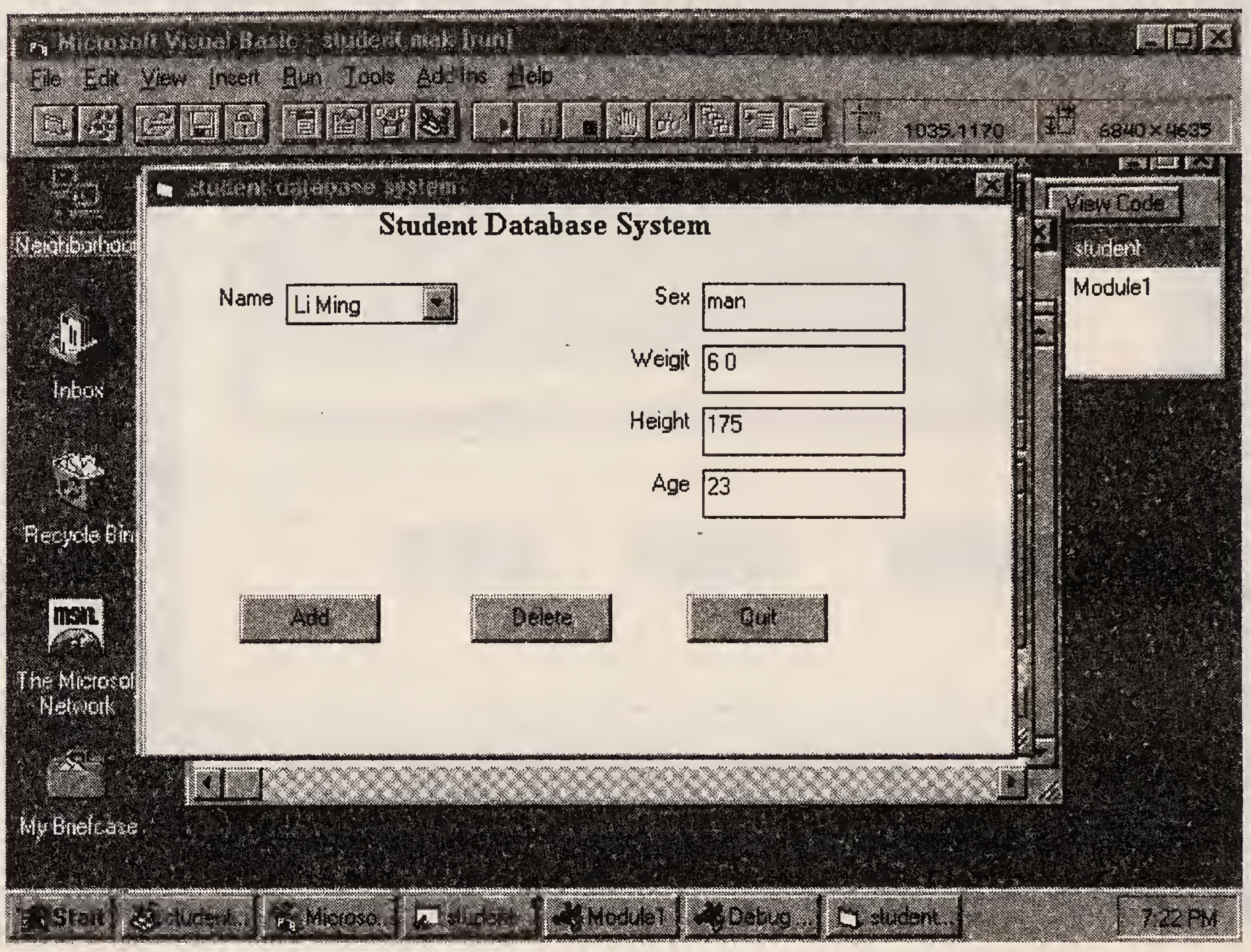
Weigjt 0

Height 0

Age 0

图10. 23　新学生的缺省设置记录

用户可以修改学生(Li Ming)记录的缺省设置。例如，用户可以在文本框中键入以下各项：man，60，175，23，见图10.24。


Microsoft Visual Basic - student.mak [run]
File Edit View Insert Run Tools Add-Ins Help
1035,1170
6840 x 4635
student database system
Student Database System
Add
Delete
Quit
View Code
student
Module1
Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Start
student...
Microso...
student
Module1
Debug...
student...
7:22 PM

Name　Li Ming

Sex　man

Weigit　6 0

Height　175

Age　23

图10.24　修改学生记录的缺省设置

用户单击 Quit 按钮结束程序的运行，这时学生 Li Ming 的记录已加入到文件中。如用户再次运行程序，可看到学生(Li Ming)的有关记录添加到文件中，如图10.25。用户单击学生时，文本框就显示有关记录。如用户想删除一个学生记录，可以单击 Delete 按钮。

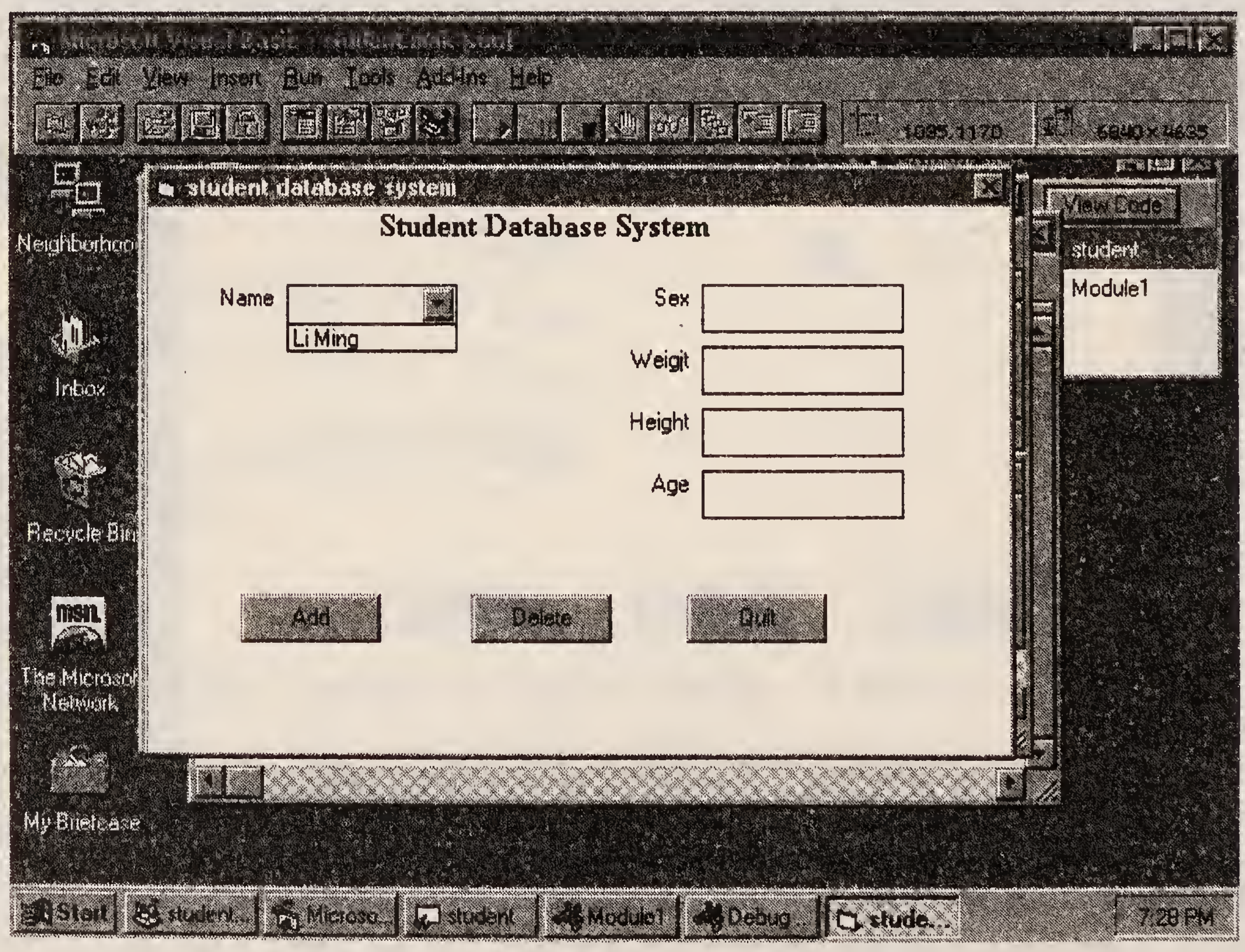

Microsoft Visual Basic - student.mak [run]
File Edit View Insert Run Tools Add-Ins Help
1035,1170
6840 x 4635
student database system
Student Database System
Name
Li Ming
Sex
Weigit
Height
Age
Add
Delete
Quit
View Code
student
Module1
Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Start
student...
Microso...
student
Module1
Debug...
stude...
7:28 PM

图10.25　添加到文件中的学生记录

10.6 二进制文件

二进制文件同随机文件一样，文件一经打开，就可以同时读写。但是它可以不必一定要组织一定长度的记录，这种存储方式最为灵活。其中文件打开及读写的格式为：

```
Open filename For Binary As # filenumber
Put #filenumber,position,variable
Get #filenumber,position,variable
```

读写格式中的 position（位置）是以 Byte 为单位，指出下一次文件要读或要写的位置。变量可是任意类型的变量。Put/Get 语句一次读写的长度等于变量 variable 的 Byte 个数据。例如，variable 为整型，Get 语句每次读入2个字节放入变量中。

在二进制存储方式中经常要用到 Seek 函数和 Seek 语句。Seek 函数是用来返回当前文件的位置，它的语法格式为：

```
Seek(filenumber)
```

参数 filenumber 表示打开文件的文件标识符。如果文件以 Random 模式打开，Seek 函数就返回下一个要读取或写入的记录号。若文件以 Binary、Output、Append、Input 模式打开，Seek 函数就返回发生下一个操作的字节位置。

Seek 语句用来在文件中设置下一次读或写的位置，它的语法格式为：

```
Seek[#]filenumber,position
```

filenumber 为打开文件的标识符。position 描述下一次读或写将要发生的位置的数值表达式。

第十一章　错误处理与程序调试

一旦程序有了一定的复杂程度，无论用户在编程工作中是如何地仔细，总是会不可避免地出现一些错误。有的错误非常简单，很容易就可以检查出来。而有些错误则隐蔽得很深，当用户经验不是很丰富时，单靠看程序代码是很难将其找出来的。这时就需要借助一些工具来对程序进行测试，从而发现并进一步改正错误。

11.1　错误类型

11.1.1　语法错误

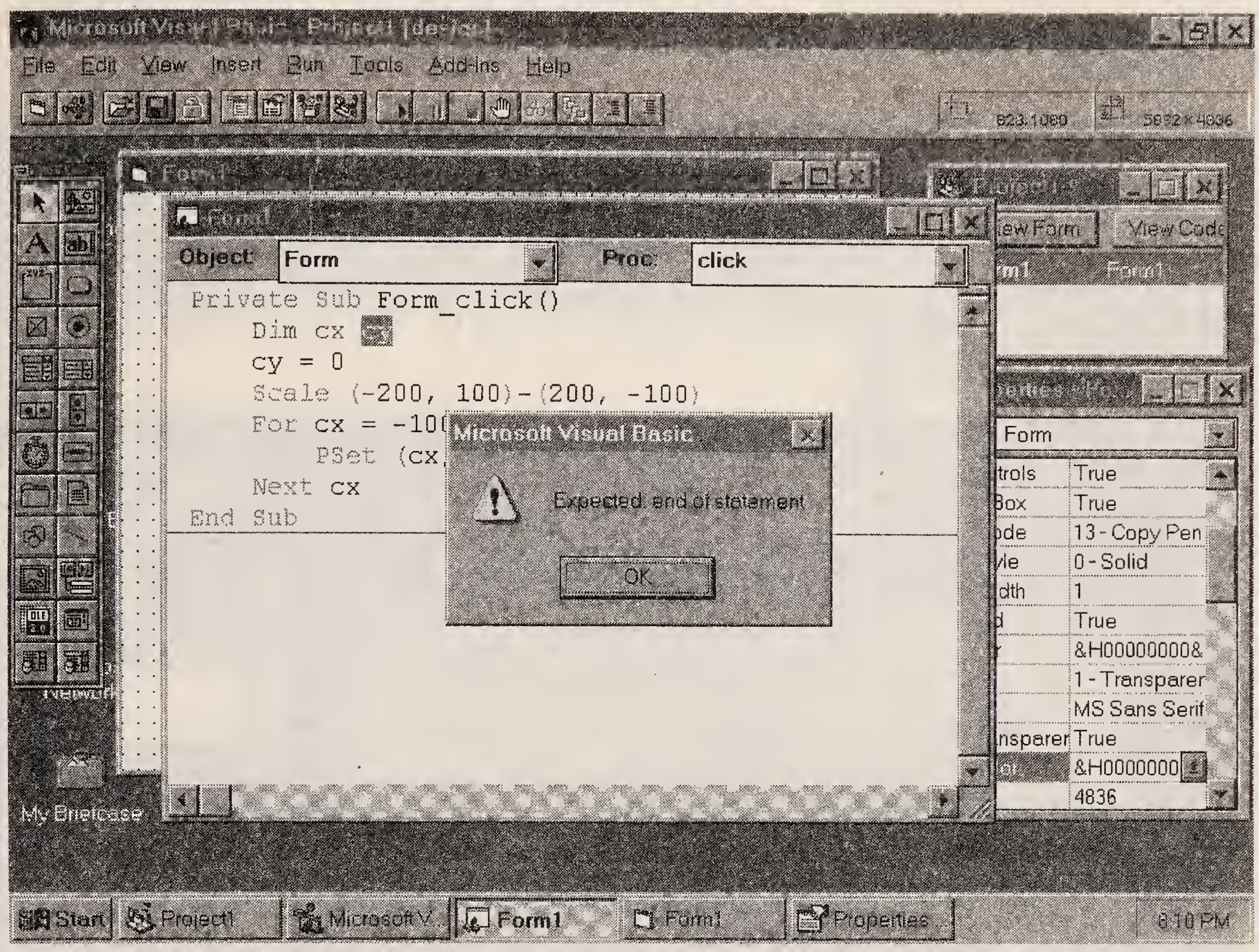
在用户编写程序的时候，Visual Basic 将对用户编写的程序进行语法检查，看看是否有语法错误。当用户出现打字错误，遗漏了某些关键字或标点符号或者出现其他的语法错误，在将光标移到其他行或按回车键来结束这一行的输入时，Visual Basic 将会检查并显示错误。例如，我们在一个 Form 的代码窗口中输入下例程序：

```
Private Sub Form _ Click()
    Dim CX CY
    CY=0
    Scale(－200,100)－(200,－100)
    For CX=－100 To 100
        PSet(CX,CY),QBColor(4)
    Next CX
End Sub
```

这段代码的意思是定义 Form 窗口新的标度，在新标度下在 Form 中央画一直线。但当输入完第一行(即 Dim CX CY)按回车时，Visual Basic 在 Form 窗口中弹出一窗口，且刚输入完的那一行中的字符被涂成了红色，出错的部分被高亮显示，如图11.1所示。

小窗口中提示的出错信息是，这一语句没有结束。原来，忘了在 CX 与 CY 之间插入“,”符号。这时，用户必须单击小窗口中的 OK 按钮，关闭这一窗口，然后对出错行进行修改。用户可以单击 Delete 键将高亮部分的字符删除掉，或将光标移到出错行，单击鼠标。这时出错行又变为正常颜色，就可进行修改了。如果用户不想对其进行修改，可以将光标移到下一行继续输入程序，但出错行中的字符依旧是红色的，直到被修改正确为止。

如果用户对于 Visual Basic 所提供的出错信息不太理解，或信息太少，可以在这时单击 F1键，即会出现关于这一出错信息的帮助文件，如图11.2所示。用户还可以在单击 OK 按钮关闭显示出错信息的小窗口后，点亮语句中的方法、函数或关键字，单击 F1键，即出现关于这一方法或函数的帮助文件。


Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
Form1
Project1
View Form
View Code
Properties
Start
Project1
Microsoft V...
Form1
Properties
8:10 PM
My Briefcase

Object: Form　Proc: click

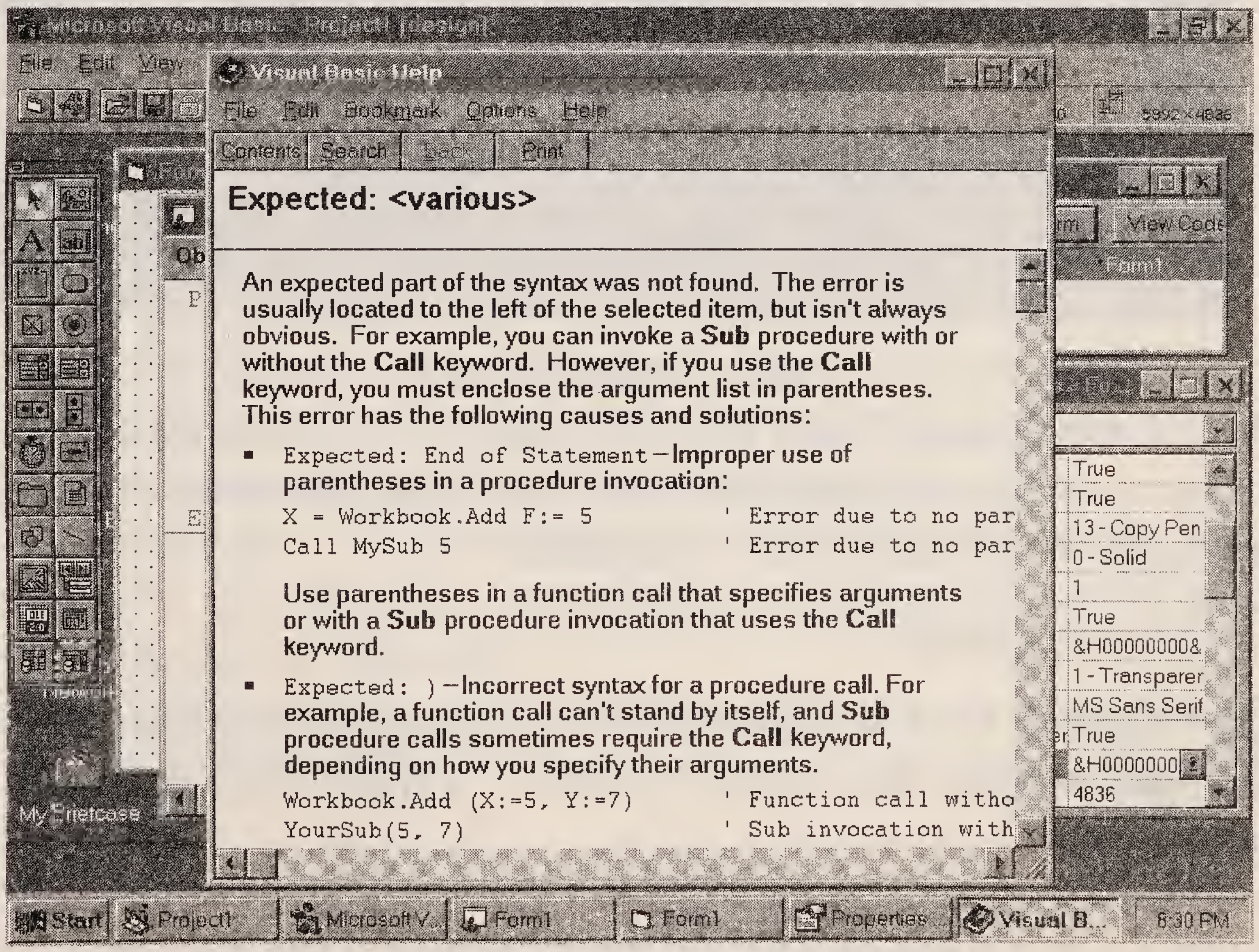
```
Private Sub Form_click()
    Dim cx cy
    cy = 0
    Scale (-200, 100)-(200, -100)
    For cx = -10(
        PSet (cx,
    Next cx
End Sub
```

Microsoft Visual Basic

Expected: end of statement

OK

图11.1　一语法错误产生的信息


Microsoft Visual Basic - Project1 [design]
File Edit View
Visual Basic Help
File Edit Bookmark Options Help
Contents
Search
Back
Print
View Code
Start
Project1
Microsoft V...
Form1
Properties
Visual B...
8:30 PM
My Briefcase

Expected: <various>

An expected part of the syntax was not found. The error is usually located to the left of the selected item, but isn't always obvious. For example, you can invoke a **Sub** procedure with or without the **Call** keyword. However, if you use the **Call** keyword, you must enclose the argument list in parentheses. This error has the following causes and solutions:

- `Expected: End of Statement`—Improper use of parentheses in a procedure invocation:

  ```
  X = Workbook.Add F:= 5           ' Error due to no par
  Call MySub 5                     ' Error due to no par
  ```

  Use parentheses in a function call that specifies arguments or with a **Sub** procedure invocation that uses the **Call** keyword.

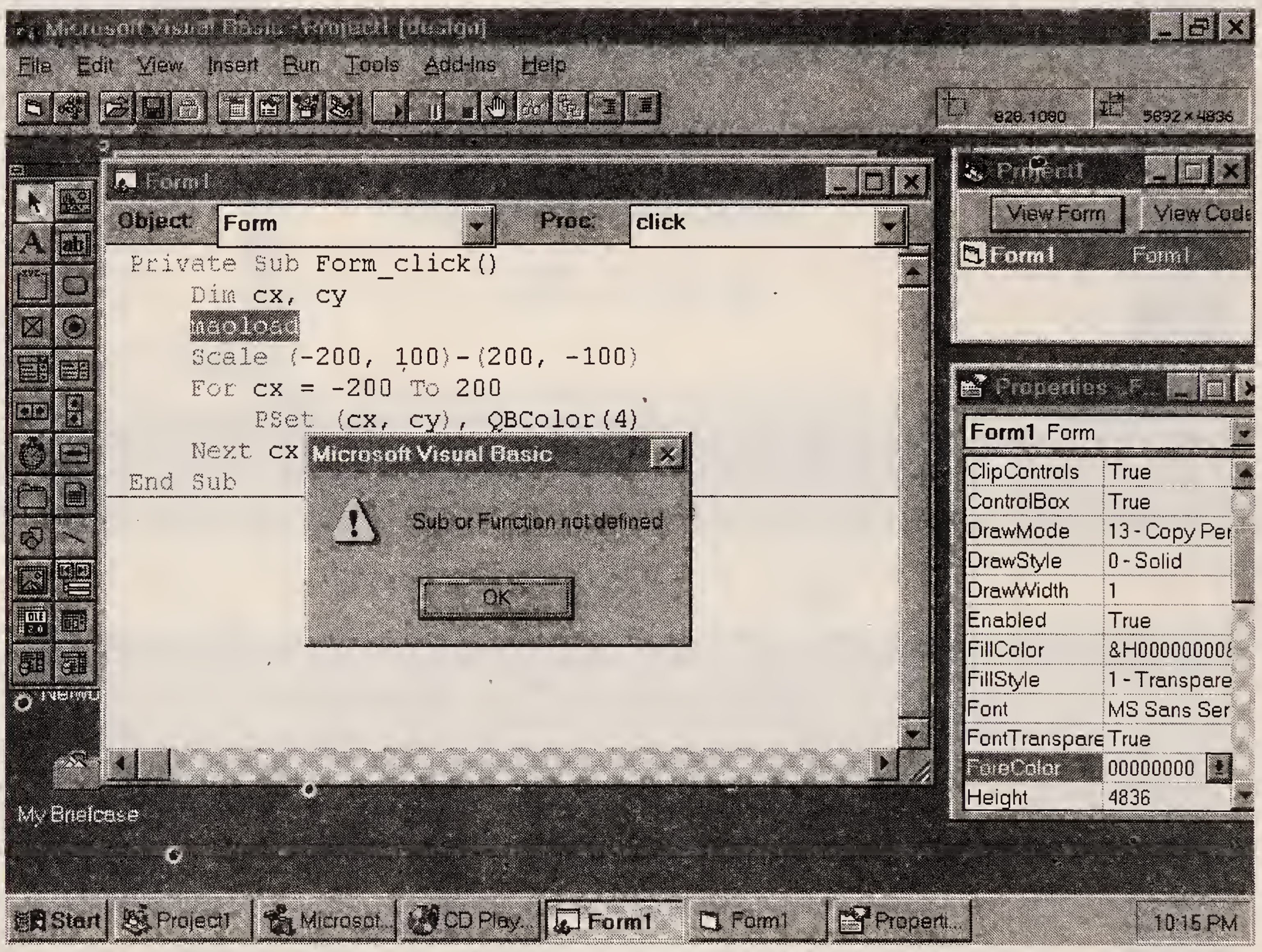
- `Expected: )`—Incorrect syntax for a procedure call. For example, a function call can't stand by itself, and **Sub** procedure calls sometimes require the **Call** keyword, depending on how you specify their arguments.

  ```
  Workbook.Add (X:=5, Y:=7)        ' Function call witho
  YourSub(5, 7)                    ' Sub invocation with
  ```

图11.2　单击F1键显现的关于错误的帮助信息

11.1.2 编译错误

编译错误是指将程序编译成 EXE(可执行文件)时,或单击 Start 命令时,由于用户未定义变量、遗漏了某些关键字等产生的不正确代码而引起的错误,这时 Visual Basic 也将弹出一小窗口,显示出错信息,并点亮出错的一整行内容,如图11.3所示。同时 Visual Basic 停止编译,并继续停留在 Code 窗口。如果用户是在 Form 窗口下对其进行编译,则发现错误后将重新回到 Code 窗口。


Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
828,1080
5892 x 4836
Form1
Object: Form
Proc: click
Project1
View Form
View Code
Form1 Form1
Properties - F...
Form1 Form
Microsoft Visual Basic
OK
My Briefcase
Start
Project1
Microsoft...
CD Play...
Form1
Form1
Properti...
10:15 PM

```
Private Sub Form_click()
    Dim cx, cy
    maoload
    Scale (-200, 100)-(200, -100)
    For cx = -200 To 200
        PSet (cx, cy), QBColor(4)
    Next cx
End Sub
```

Sub or Function not defined

ClipControls	True
ControlBox	True
DrawMode	13 - Copy Per
DrawStyle	0 - Solid
DrawWidth	1
Enabled	True
FillColor	&H00000000
FillStyle	1 - Transpare
Font	MS Sans Ser
FontTranspare	True
ForeColor	00000000
Height	4836

图11.3 编译时产生的错误

在图11.3中出现的错误,是由于我们在程序中加入了一 maoload 语句,由于用户没有对其进行说明,Visual Basic 对其不能辨认,因此产生编译错误。之所以在编写程序时没有被语法检查给检查出来,是因为它并不是一个语法错误。

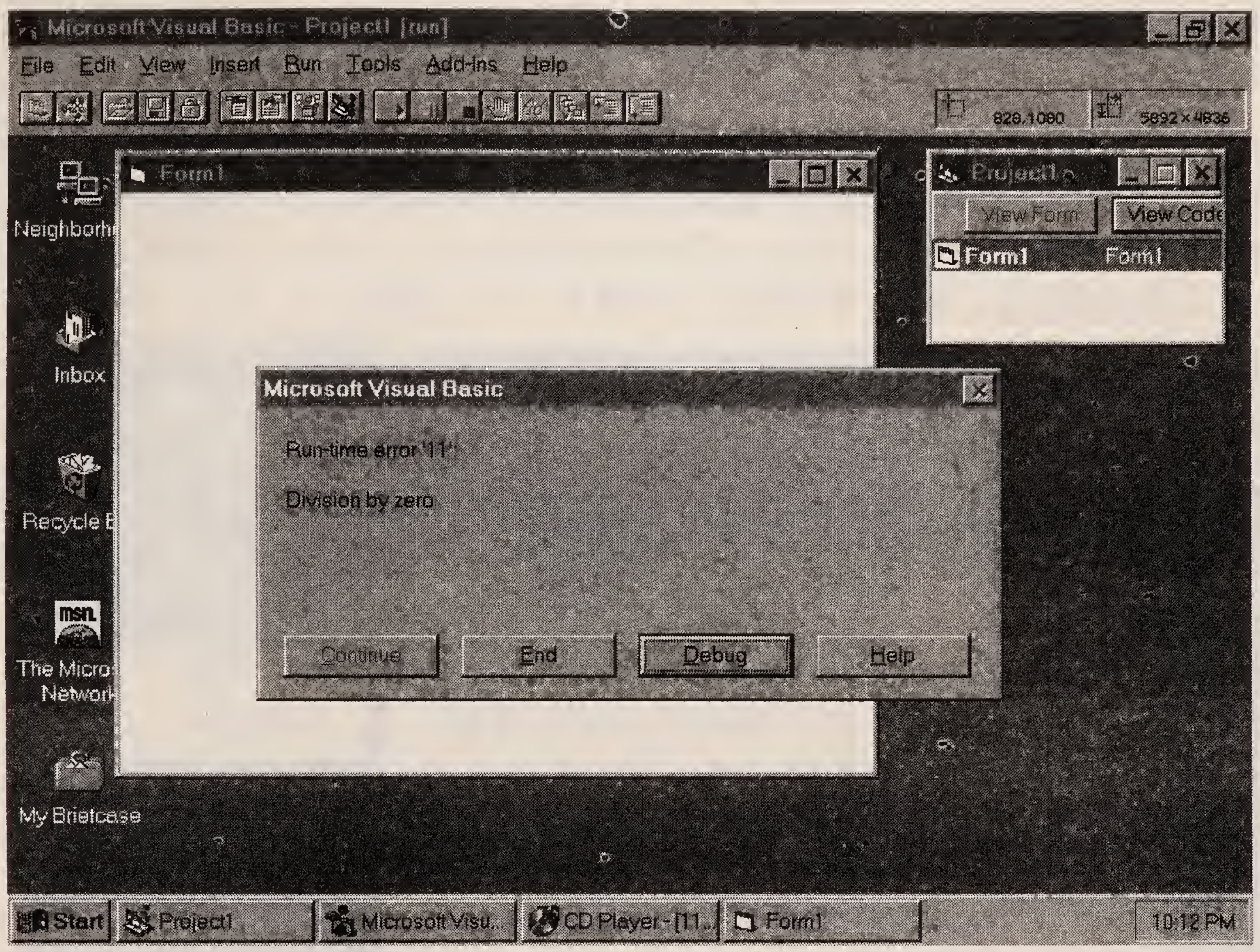
11.1.3 运行时发生的错误

运行时错误是在 Visual Basic 运行代码时发生的错误,这错误往往是由于某一语句试图执行一非法操作所造成的。例如,我们对前一程序作一适当修改。将 CY 赋值为1,即 CY=1,将 PSet 语句改为:

```
PSet(CX/CY,CY),QBColor(4)
```

这时单击工具栏中的 Start 按钮,然后单击 Form 窗口,我们会看到其结果与刚才的程序的结果几乎是一样的,只是直线向上平移了一点(当然,移动是微小的,几乎看不出来)。那么,如果这时我们在输入时忘了为 CY 赋值,会出现什么结果呢?将 CY=1行删去,这时再单击工具栏中的 Start 按钮。当单击 Form 窗口时,会弹出一窗口,如图11.4所示。显示出错原因,原来由于 CY 未被赋值,在执行时,有可能被赋零值,因而产生错误。在窗口的下方有四

个按钮，分别是 Continue（继续执行），End（结束运行），Debug（进入调试窗口），Help（显示出错信息的帮助内容）。


Microsoft Visual Basic - Project1 [run]
File Edit View Insert Run Tools Add-Ins Help
828,1080
5892×4836
Form1
Project1
View Form
View Code
Form1 Form1
Neighborho
Inbox
Recycle B
The Micro Network
My Briefcase
Microsoft Visual Basic
Continue
End
Debug
Help
Start
Project1
Microsoft Visu...
CD Player - [11...
Form1
10:12 PM

Run-time error '11':

Division by zero

图11.4　由于未赋值产生的运行错误

为了避免在运行时产生错误，用户在编写代码后，一定要检查一下，看是否有变量未被赋值，或有没有非法操作。

11.1.4　逻辑错误

当计算机按照程序的要求运行，但得到的结果却不是用户所期望得到的，这就说明发生了逻辑错误。逻辑错误往往是由于用户在设计程序时存在着逻辑缺陷造成的。例如，使用了错误的变量，定义了错误的变量类型，产生无休止的循环以及数组错误等等。

有的时候，错误很明显，可以很容易的找到它。而在大多数情况下，错误都很难找到，因为错误的原因与产生错误结果的语句之间可能隔得很远。逻辑错误甚至不会产生错误信息，但它会使程序计算和输出不正确的结果。

例如，在第九章的9.7.4节的动画设计中所举的地球图形在 Form 窗口内来回不停的运动，且碰到边界会发生碰撞现象的例子中。刚开始，对 Timer1_ Timer()事件过程的代码是这样编写的：

```
Private Sub Timer1_ Timer()
    Image1. Move Image1. Left + CX,Image1. Top + CY
    If Image1. Left + Image1. Width > ScaleWidth + ScaleLeft
          Then CX=-50
    If Image1. Top + Image1. Height > ScaleHeight + ScaleTop
          Then CY=-75
    If Image1. Top=0 Then CY=50
    If Image1. Left=0 Then CX=75
End Sub
```

本意是当图形运行到 Form 窗口底边和右边界时分别转向上或左运动，当到达 Form 窗口最左边及最上边时，将分别向右及向下运动。结果当运行程序时，我们发现，当图形运动到 Form 窗口左边界或上边界时，它穿过窗口出去了，如图11.5所示。经过多次查找，才发现原来错误出现在 If Image1. Top=0 Then CY=50和 If Image1. Left=0 Then CX=75语句上。因为 Visual Basic 很难准确判断 Image1. Top=0及 Image1. Left=0，因而无法执行这一语句，这就是一个逻辑错误，只要将等号改为小于等于(<=)即可。

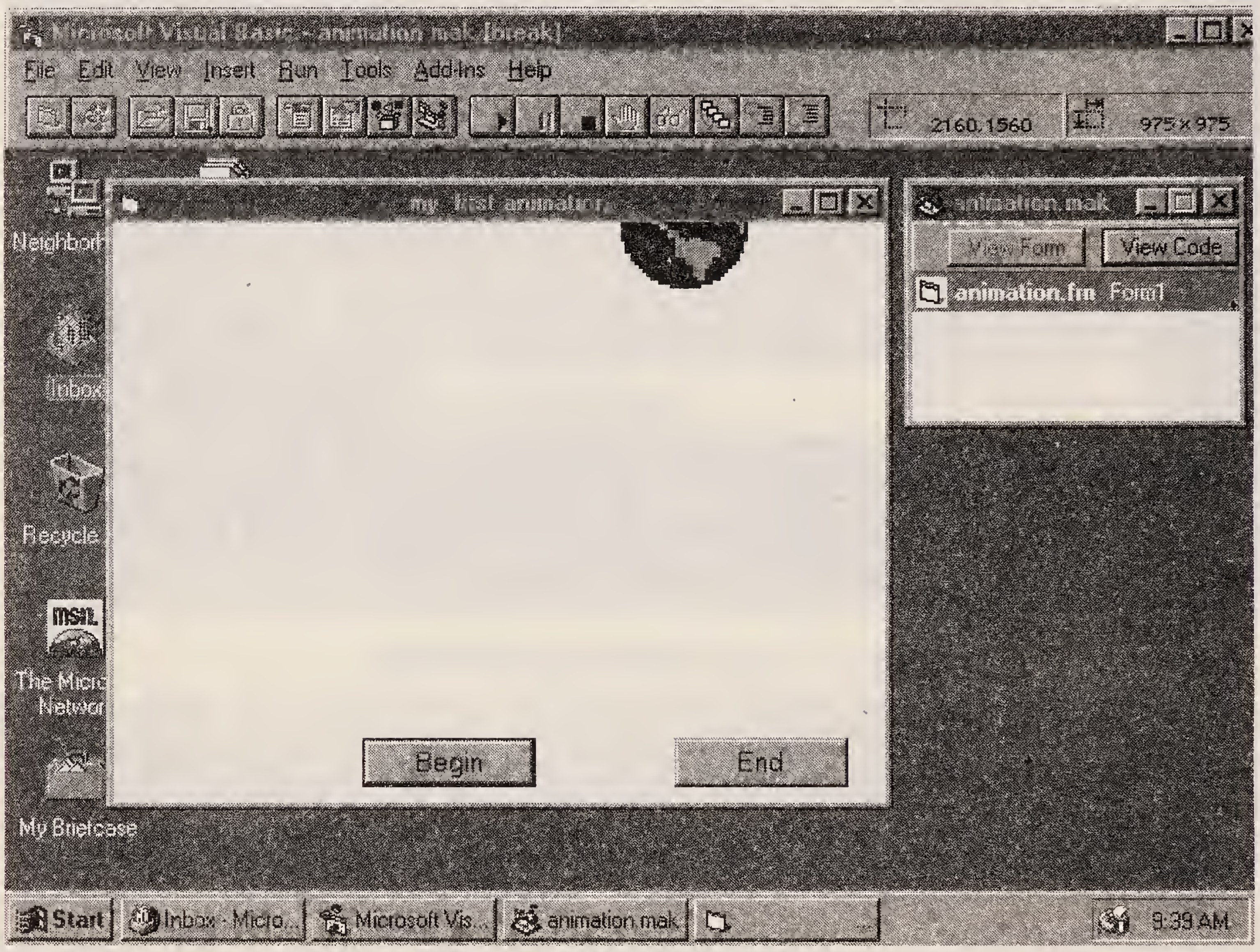

Microsoft Visual Basic - animation.mak [break]
File Edit View Insert Run Tools Add-Ins Help
2160,1560
975x975
Neighborh
Inbox
Recycle
The Micro Networ
My Briefcase
my first animation
Begin
End
animation.mak
View Form
View Code
animation.frm Form1
Start
Inbox - Micro...
Microsoft Vis...
animation.mak
9:39 AM

图11.5　一逻辑错误引起的画面

11.2　错误处理

想必用户在使用 Visual Basic 时已经遇到过运行时出现的错误了吧?运行时产生的错误，甚至引起程序崩溃的错误都是我们不愿碰到的。运行时的错误常常是由于某些语句在执行时，由于某种原因而产生的，这并不是 Visual Basic 不够坚固而不能发现这些错误，而是由于用户没有告诉编译器在发生错误时该怎么做。

幸运的是，我们并不是对这些错误束手无策。Visual Basic 允许用户使用错误处理方法来对错误进行处理。所谓的错误处理，就是允许用户所编写的程序对错误进行响应。当一个错误发生了，程序应捕捉到这一错误，并知道怎样做来处理这一错误。当用户觉得程序中某一语句可能会发生错误时，就将错误处理程序放在同一程序中。在某些场合中，程序需要重新执行产生错误的指令；有的时候程序需要显示一条错误信息；更有一些场合，程序执行下一条指令而忽略该错误。在程序中增加指令来捕捉和处理错误的步骤是：

（1）设置一个错误陷阱(trap)。即在程序中合适的地方增加一些语句来告诉程序在发生错误时该怎样操作。

（2）编写错误处理程序。当程序中出现错误时，程序的运行将转到错误处理程序，错误处理程序将根据所发生错误的类型决定采取何种措施。

（3）退出错误处理程序。在错误处理程序决定采取什么措施之后，应初始化某动作以纠正错误。为了达到这一点，错误处理程序不得不包括一些退出处理程序及初始化正确动作的代码。

11.2.1　使用错误处理程序

用户可以在怀疑可能出现错误的程序中添加错误处理程序。一般来说，错误处理经常用来检测那些对程序会产生影响的外部事件。例如，同用户联网的微机的驱动器错误，没有关闭软盘驱动器的小门，或者打印机缺纸等等。

下表列出的是用错误处理程序经常检测的错误：

错误	描述
软盘错误	软盘未格式化，驱动器小门未关，或是一坏磁盘
打印机错误	未连接打印机，缺纸或设置错误
溢出错误	一个 Form 窗口的内容过多
剪贴板错误	数据传输错误或剪贴板错误
逻辑错误	不能被编译器检测的语法和逻辑错误

11.2.2　设置错误陷阱(trap)：使用 On Error 语句

Visual Basic 提供给用户 On Error 语句来检测错误。用户可以使程序继续执行而忽略该错误，也可以调用错误处理程序，或者关闭错误处理。执行这些不同功能的 On Error 语句分别为：

```
On Error Resume Next
On Error GoTo Linelable
On Error GoTo 0
```

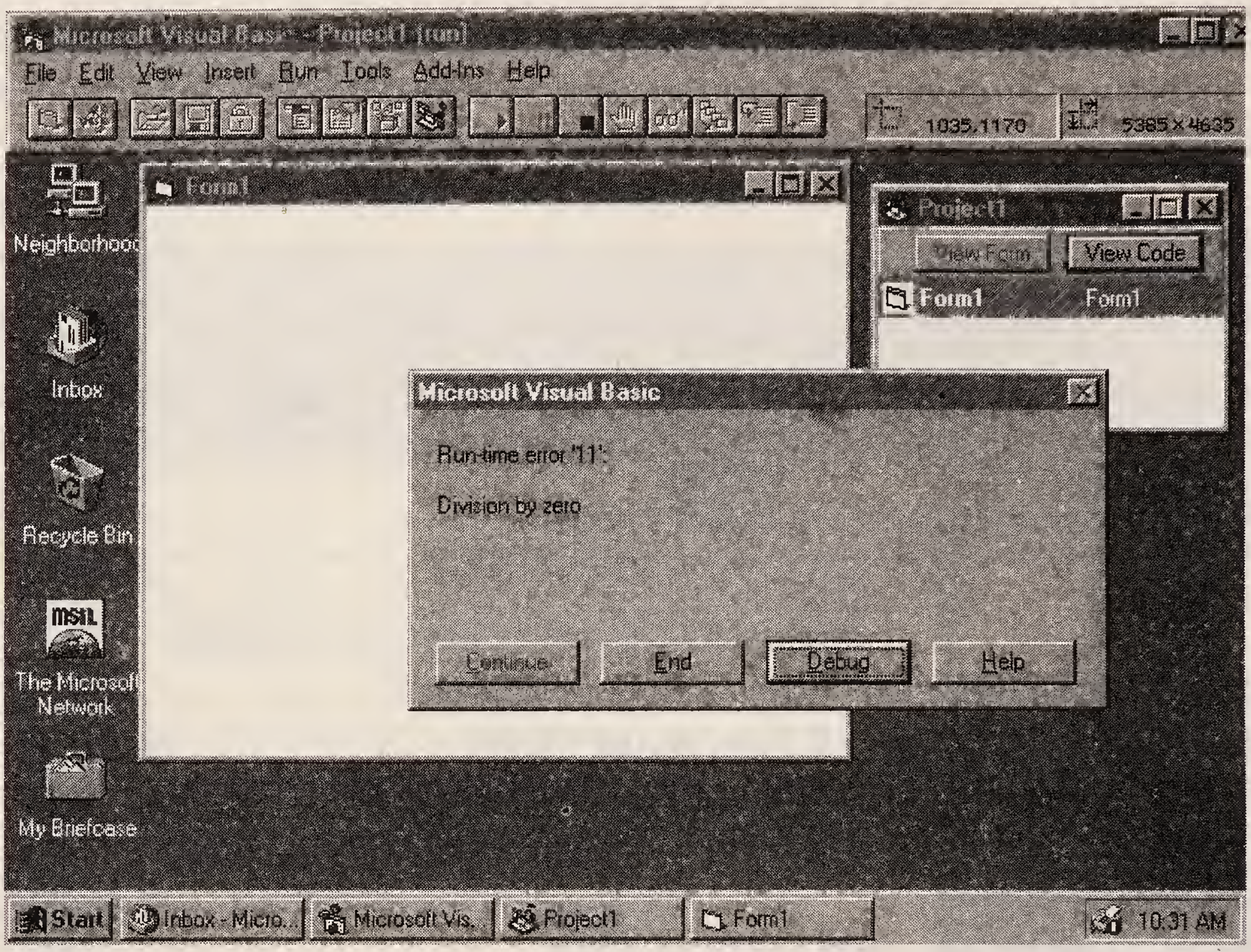
On Error Resume Next 语句将忽略错误行，即遇到错误行时，并不执行这一行语句，而是越过它继续执行下一语句。例如：

在 File 菜单中单击 New Project 命令，将 Form 窗口的 Caption 属性设置为 Test "Resume Next" Statement。然后双击 Form 窗口或单击 Project 窗口中的 View Code 按钮，在打开的 Code 窗口中输入下列代码：

```
Private Sub Form _ Click()
    Dim X,Y
    For X=0 To 9
        Y=70 Mod X
    Next X
    Print Y
End Sub
```

然后单击工具栏中的 Start 按钮运行程序。单击 Form 窗口，这时会弹出一对话框，告知

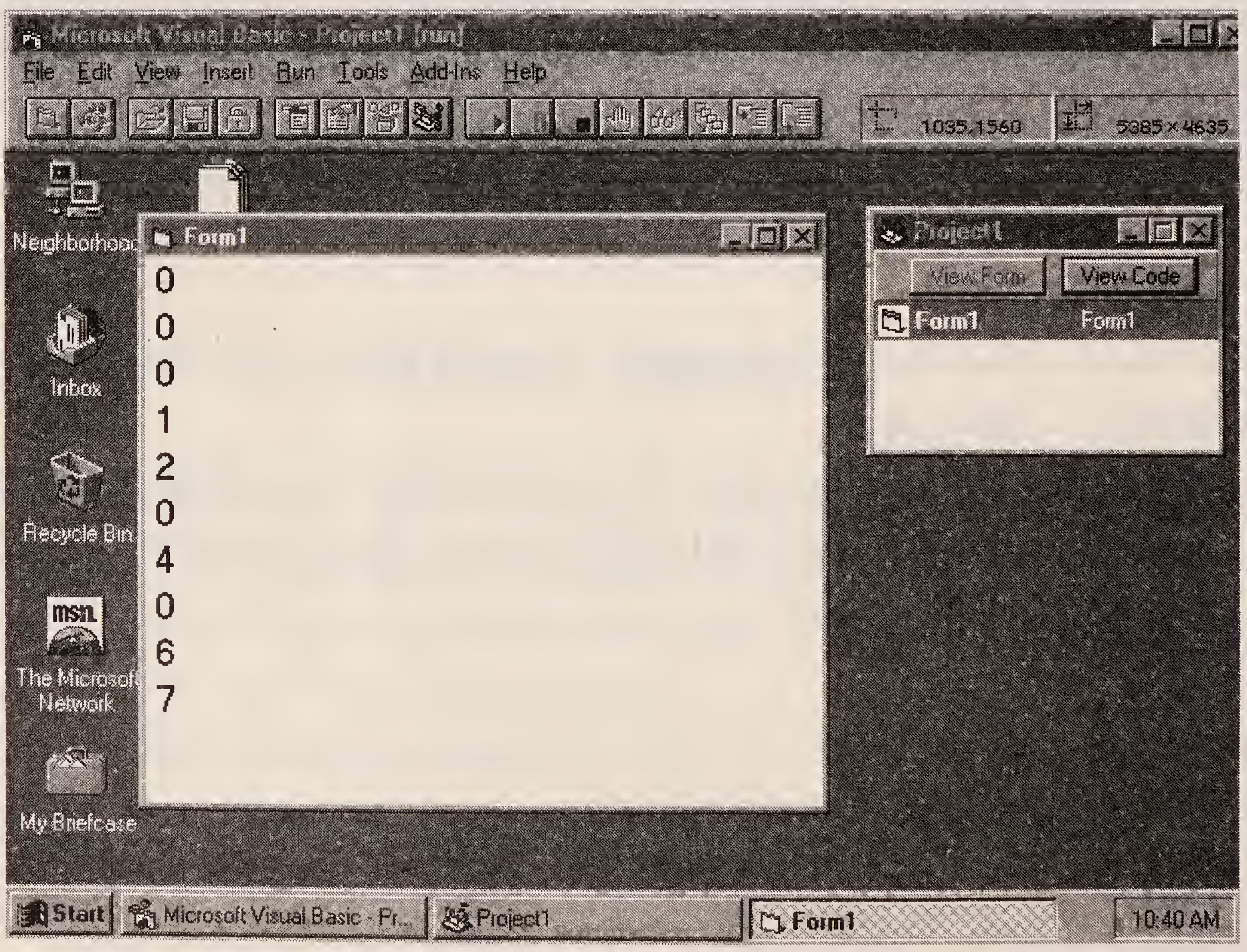
用户发生错误，如图11.6所示。单击工具栏中的 End 按钮结束程序的运行，回到 Code 窗口，在 Form _ Click()事件的过程头下加入一句“On Error Resume Next”语句。再单击工栏中的 Start 按钮运行程序，这时单击 Form 窗口，即出现结果。如图11.7所示。这是因为当 X=0时 Visual Basic 按照“On Error Resume Next”指令，略过这一错误执行下一语句。


Microsoft Visual Basic - Project1 [run]
File Edit View Insert Run Tools Add-Ins Help
1035,1170
5385 x 4635
Form1
Project1
View Form
View Code
Form1
Form1
Microsoft Visual Basic
Continue
End
Debug
Help
Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Start
Inbox - Micro...
Microsoft Vis...
Project1
Form1
10:31 AM

Run-time error '11':

Division by zero

图11.6　由于除零所产生的错误


Microsoft Visual Basic - Project1 [run]
File Edit View Insert Run Tools Add-Ins Help
1035,1560
5385 x 4635
Form1
Project1
View Form
View Code
Form1
Form1
Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Start
Microsoft Visual Basic - Pr...
Project1
Form1
10:40 AM

```
0
0
0
1
2
0
4
0
6
7
```

图11.7　使用“Resume Next”语句忽略错误

11.2.3　使用 On Error GoTo 语句

错误处理程序就是放在程序中用以处理程序中出现错误的一个代码块。当错误发生时，能使程序跳到错误处理程序。On Error GoTo 语句的使用格式为：

```
On Error GoTo Label
```

在这里 Label 代表错误处理程序所在位置的行标。错误处理程序一般都处于程序的最底部，即某一事件过程结束的位置。不同的错误处理程序有不同的行标，且行标号必须跟一冒号（：），如：

```
ErrorHandler：
```

当执行完错误处理程序后，使用 Exit Sub 语句来退出程序。下面就举一个小例子，来看看怎样使用 On Error GoTo 语句。

在这个例子中，我们将使用代码往 Form 窗口中装入一幅图画，如果这段图画并不存在，则弹出对话框，告知用户，此图画并不存在。

（1）在 File 菜单中选择 New Project 命令，将 Form 窗口的 Caption 属性设置为 Error Test。

（2）单击 Project 窗口的 View Code 按钮。在出现的 Code 窗口的 Object 列表框中选择 Form，在 Procedure 列表框中选择 Click，这时即在 Code 窗口中出现 Form _ Click 事件的过程头与过程尾。

（3）在事件过程的第一列输入下列代码：

```
On Error GoTo FileError
```

这句代码表示在过程中设置了错误处理，且错误处理程序的行标为 FileError，当程序发生错误时即转去执行 FileError 所要求的操作。

（4）然后在上一行代码下输入下列代码：

```
Form1. Picture=LoadPicture(″C：\metafile\business\laptop. wmf″)
Exit Sub
```

这段代码表示将从 C 盘中的 metafile\business 文件中装入元文件 laptop. wmf 图形到 Form 窗口中。安装图形代码下的 Exit Sub 语句用于表示当程序没有错误时即退出程序，不用去执行错误处理程序了。

（5）在装入图形的语句下面，就可以编写错误处理程序了：

```
FileError：
    MsgBox(″I can't find laptop1. wmf in C：\metafile\business),,
     ″File Not Found″
    Exit Sub
```

这段代码的意思是当 Visual Basic 发现 C 盘中并没有 laptop1. wmf 这一文件时，即弹出对话框，告知用户文件找不到，且退出程序的执行。

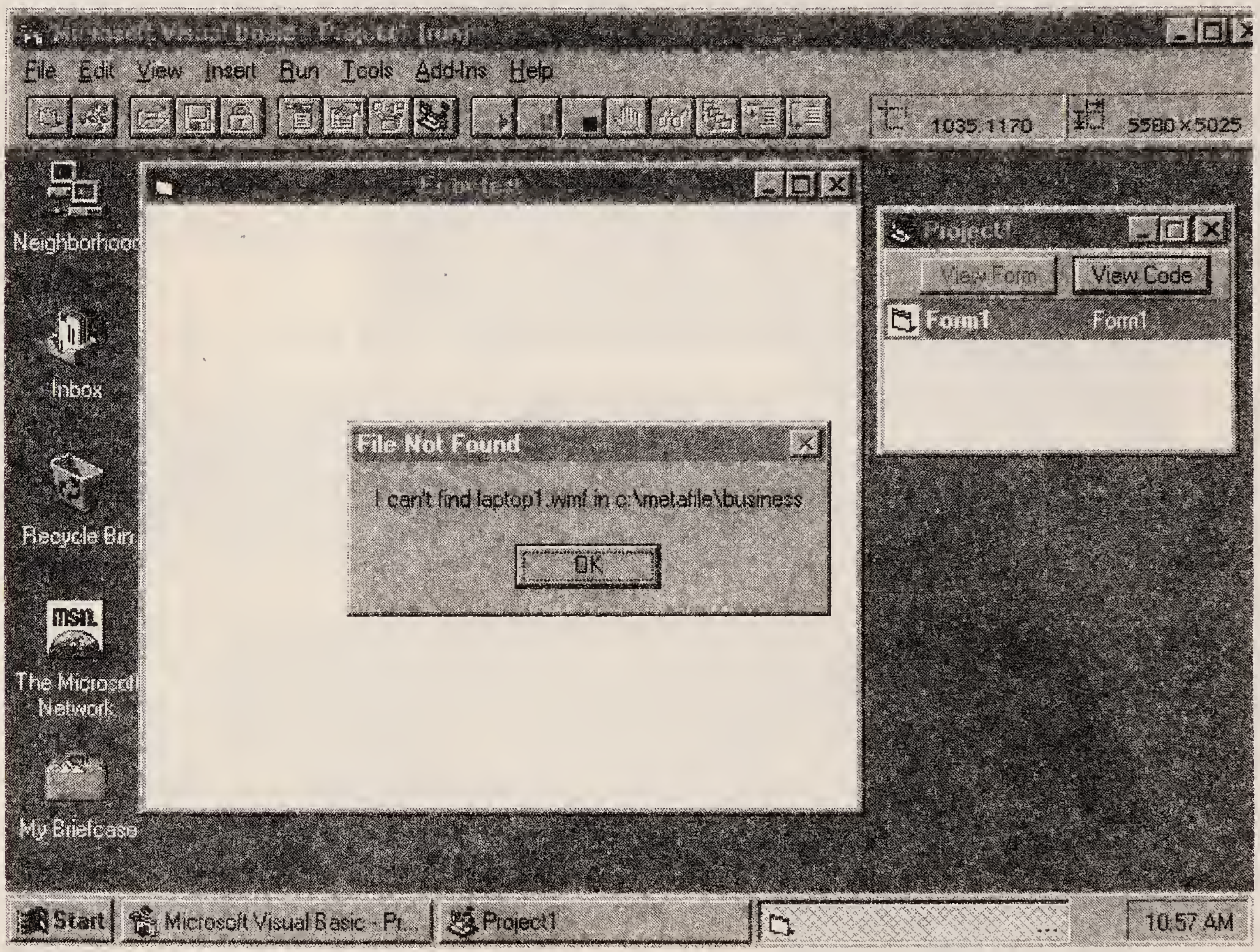
（6）这样，一个具有错误处理程序的程序就完成了。整个程序的过程代码为：

```
Private Sub Form _ Click()
    On Error GoTo FileError
    Form1. Picture=LoadPicture(″C：\METAFILE\BUSINESS\LAPTOP. WMF″)
    Exit Sub
```

```
FileError:
    MsgBox("I can't find laptop.wmf in C:\metafile\business"),,
     "File Not Found"
    Exit Sub
End Sub
```

(7) 单击菜单条中的 File 菜单，单击 Save Project 命令，在 Save File 对话框中输入 ERRORTEST.FRM 文件名。在随后出现的 Save Project 对话框中将工程文件命名为 ERRORTEST.MAK，将文件保存起来。

(8) 将文件存盘后，这时可以运行程序。单击工具栏中的 Start 命令，在运行时出现的 Form 窗口中单击鼠标，即弹出一对话框，如图11.8所示。


File
Edit
View
Insert
Run
Tools
Add-Ins
Help
1035,1170
5580×5025
Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Project1
View Form
View Code
Form1
Form1
File Not Found
OK
Start
Microsoft Visual Basic - Pr...
Project1
10:57 AM

I can't find laptop1.wmf in c:\metafile\business

图11.8 由于运行错误产生的对话框

(9) 单击菜单条中的 End 按钮，结束程序的运行。这时，单击 Project 窗口中的 View-Code 按钮，对程序代码进行改正。将 Form1.Picture=LoadPicture("C:\METAFILE\BUSINESS\LAPTOP1.WMF")改为：

```
Form1.Picture=LoadPicture("C:\VB\METAFILE\BUSINESS\LAPTOP1.WMF")
```

(10) 关闭 Code 窗口，运行程序，单击 Form 窗口，这时即在 Form 窗口中出现图形 LAPTOP1.WMF。如图11.9所示。

一个错误陷阱是被激活的，一旦其被激活，任何运行错误将导致程序控制跳至它，并执行其所编制的语句。错误处理程序从捕捉到运行错误开始到遇到其中的 Resume，Exit Sub 或 Exit Function 语句期间都是活动的。例如在前面那个 ErrorTest 例子中，当程序发生错误时，将转去执行 FileError 错误处理程序，这时 FileError 错误处理程序将被激活，直到遇到 Exit Sub 退出 FileError 错误处理程序为止，这一程序都是活动的。

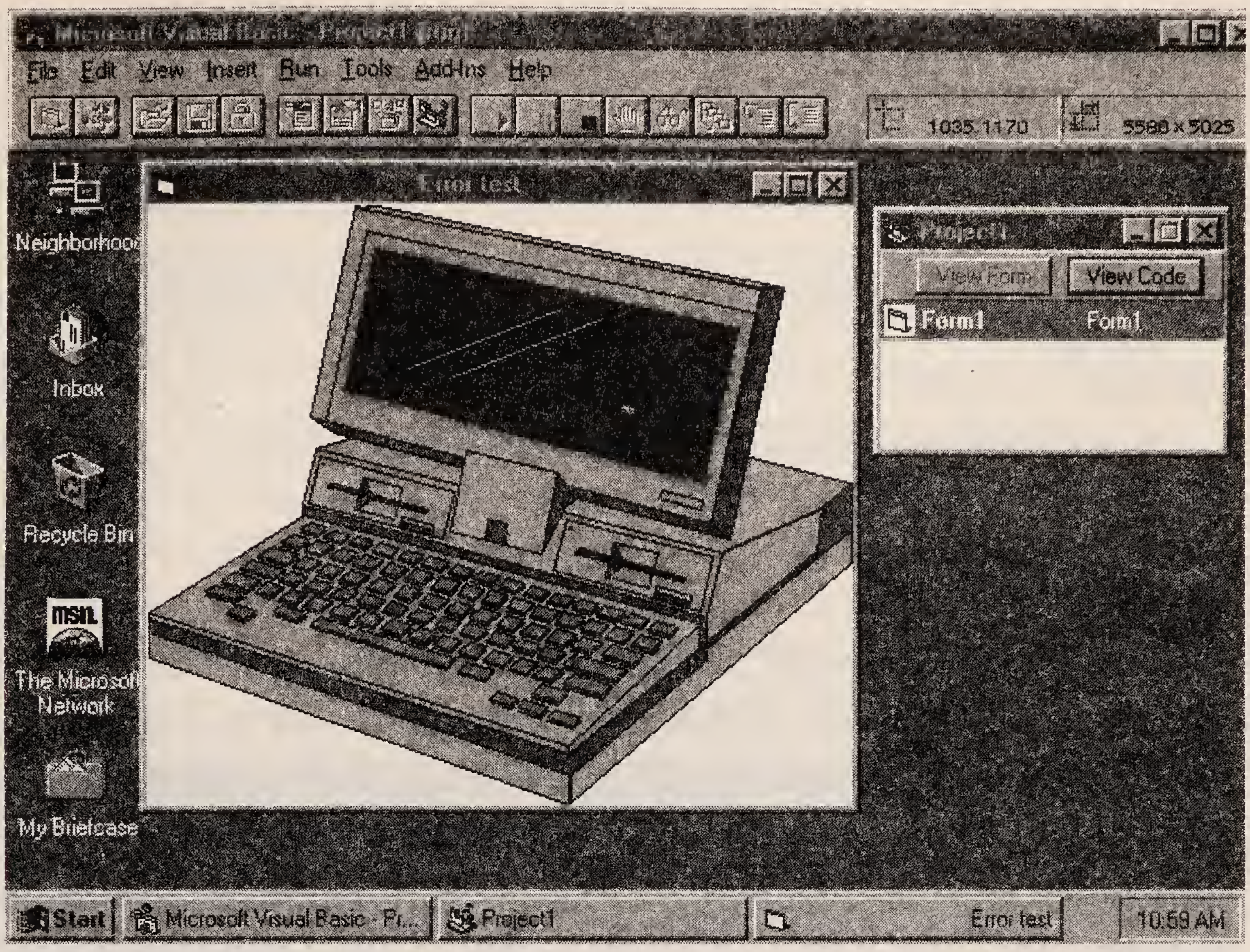

Microsoft Visual Basic - Project1 [run]
File Edit View Insert Run Tools Add-Ins Help
1035,1170
5588 x 5025
Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Error test
Project1
View Form
View Code
Form1
Form1
Start
Microsoft Visual Basic - Pr...
Project1
Error test
10:59 AM

图11.9　在 Form 窗口中安装 LAPTOP. WMF 图形

当 Visual Basic 执行完错误处理程序后就要退出，继续执行程序的其他语句。在上一个例子中，使用了 Exit Sub 语句，终止错误处理程序，转而恢复执行外部过程和函数，即执行 End Sub 语句，结束了一个程序的运行。除了使用 Exit Sub 退出错误处理程序外，还可以使用下列语句。

Resume。使得能够重新执行产生错误的那条语句。当用户已经修改了错误，并希望重新运行语句时，可在错误处理程序中使用 Resume 语句。例如，当用户没有将软盘装入软盘驱动器，而执行从磁盘中读取信息的程序时，将会产生错误。这时用户可以设立错误处理程序，提醒用户插入磁盘，并以 Resume 语句结尾。这样，当发生错误时，用户只要插入软盘，即可重新执行从软盘中读取信息的程序。

Resume Next。使得错误发生以后，当错误处理程序已经解决了错误或用户认为错误无关紧要时，接着执行错误行的下一语句。例如，将前面 Error Test 例子中 FileError 错误处理程序中的 Exit Sub 语句改为 Resume Next 语句。然后在 Form1. Picture＝LoadPicture 语句后面添加一句：

```
Form1. Picture=LoadPicture(C:\VB\METAFILE\BUSINESS\LAPTOP1. WMF)
```

这样，当运行程序时，我们会发现，当单击 Form 窗口时，弹出一对话框“File Not Found”，当单击对话框中的 OK 按钮后，对话框即消失。并接着在 Form 窗口上安装了 LAPTOP1. WMF 图形。

Resume Label。这一语句使得重新执行该行标所指定的语句。如同错误处理程序一样，由行标所定的行必须在同一过程中。例如，将 Error Test 例子中的错误处理程序中的 Exit Sub 语句改为 Resume Test。然后关闭 Code 窗口，在 Form 窗口中画一图片框。再打开 Code 窗口，将程序代码改为如下形式。

```
Private Sub Form _ Click()
    On Error GoTo FileError
    Form1. Picture=LoadPicture(″C:\METAFILE\BUSINESS\LAPTOP1. WMF″)
    Exit Sub
Test:
    Picture1. Picture=LoadPicture(″C:\VB\METAFILE\BUSINESS\3DLRSIGN. WMF″)
    Exit Sub
FileError:
    MsgBox(″I can’t find laptop1. wmf in C:\METAFILE\BUSINESS″),,
      ″File Not Found″
    Resume Test
End Sub
```

然后运行程序，在单击 Form 窗口后，窗口中出现一对话框。单击对话框中的 OK 按钮，对话框消失，随后在图片框中出现一个图形3DLRSIGN. WMF。如图11.10所示。

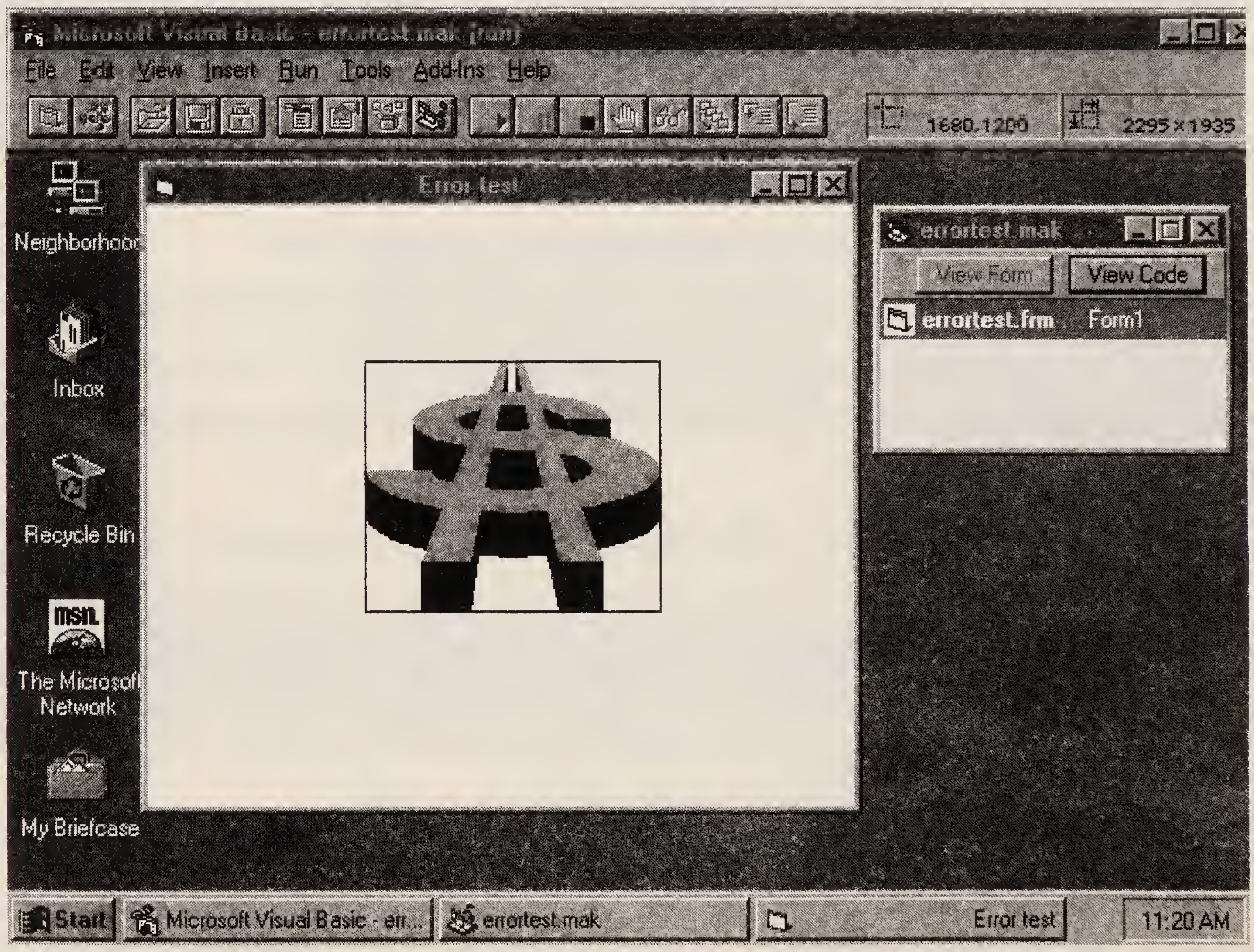

Microsoft Visual Basic - errortest.mak [run]
File Edit View Insert Run Tools Add-Ins Help
1680,1200
2295×1935
Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Error test
errortest.mak
View Form
View Code
errortest.frm
Form1
Start
Microsoft Visual Basic - err...
errortest.mak
Error test
11:20 AM

图11.10 由 Resume Label 语句来结束错误处理而出现的画面

当用户不想使用错误处理程序时，可以使用 On Error GoTo 0语句。它并不是指明第0行为出错处理代码的开始处，即使在过程中有第0行号。

11.2.4 使用 Error 和 Error $ 函数

Error 和 Error $ 函数用于返回错误代码对应的错误信息。

Error 函数的格式为：

    Error [errorcode]

Error $ 函数的格式为：

Error $［errorcode］

errorcode 为错误代码，可以省略。当省略时，返回最近运行错误对应的字符串信息。错误代码必须是1～32767之间的整数，若超过这一范围，即没有在 Visual Basic 中被定义，则返回信息“User-defined error”。

Error 函数将返回变体型，而 Error $ 返回字符串。在错误发生时，错误信息字符串中出现与错误相匹配的文字。如果使用 Error[$]（即 Error 或 Error $ ）函数时没有使用错误代码，则它将返回最后的运行错误相对应的信息串。如果发生了错误，返回的信息串中包含了与错误相对应的内容，否则 Error[$]返回一长度为0的字符串"",即空串。

下面将举一例子，发生错误时，将在对话框中显示相关的错误内容。

（1）选择 File 菜单中的 Open Project 命令，在 FileName 对话框中输入 C：\VB\ERRORTEST. MAK，单击 OK 按钮，即打开了 ErrorTest 工程文件。

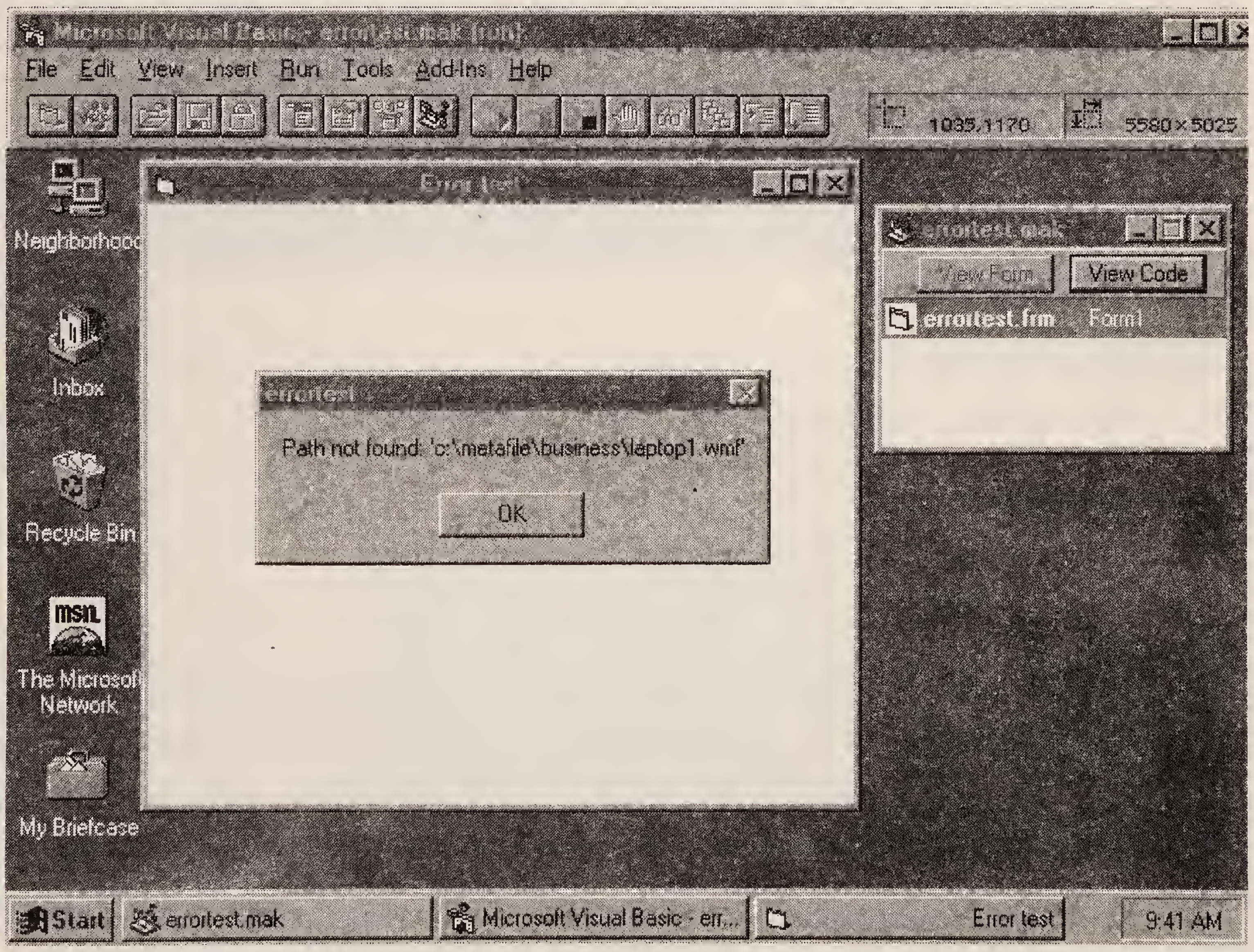
（2）单击 Project 窗口中的 View Form 按钮，使出现 Form 窗口。然后单击 View Code 按钮，在随后出现的 Code 窗口中将 fileerror 下的代码做如下改动：

```
FileError:
            MsgBox(Error)
            Exit Sub
```

（3）关闭 Code 窗口。

（4）单击工具栏中的 Start 按钮，运行程序。

（5）单击 Form 窗口。这时即出现一对话框，显示所出现错误的具体信息，“Path not found：C：\metafile\business\laptop1. wmf”，如图11. 11所示。


Microsoft Visual Basic - errortest.mak [run]
File
Edit
View
Insert
Run
Tools
Add-Ins
Help
1035,1170
5580×5025
Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Error test
errortest.mak
View Form
View Code
errortest.frm
Form1
errortest
OK
Start
errortest.mak
Microsoft Visual Basic - err...
Error test
9:41 AM

Path not found: 'c:\metafile\business\laptop1.wmf'

图11. 11　由 Error 函数返回的错误信息

（6）单击对话框中的 OK 按钮，对话框消失。然后单击工具栏中的 End 按钮，结束程序

的运行，并回到 Code 窗口。

(7) 按对话框所提示的错误信息对程序进行修改。

11.2.5 使用 Error 语句

Error 语句用来模拟错误的产生，可用来产生自定义错误。

Error 语句的格式为：

Error errorcode

errorcode 为错误代码，必须是1～32767之间的整数。若 errorcode 是由 Visual Basic 定义的错误代码，则 Error 语句模拟错误的产生，即把 Err 值设置给 errorcode。

如果用户想使用自定义错误，可以使用大于32767的整数为 errorcode 赋值。若在没有激活的错误处理程序中执行 Error 语句，Visual Basic 将显示错误信息，并且停止程序的执行。若 Error 语句指定的错误代码没有在 Visual Basic 中被定义，则显示 User-defined error 信息。下面举一个例子。

(1) 选择 File 菜单中的 New Project 命令，将 Form 窗口的 Caption 属性设置为“errorcode information”。

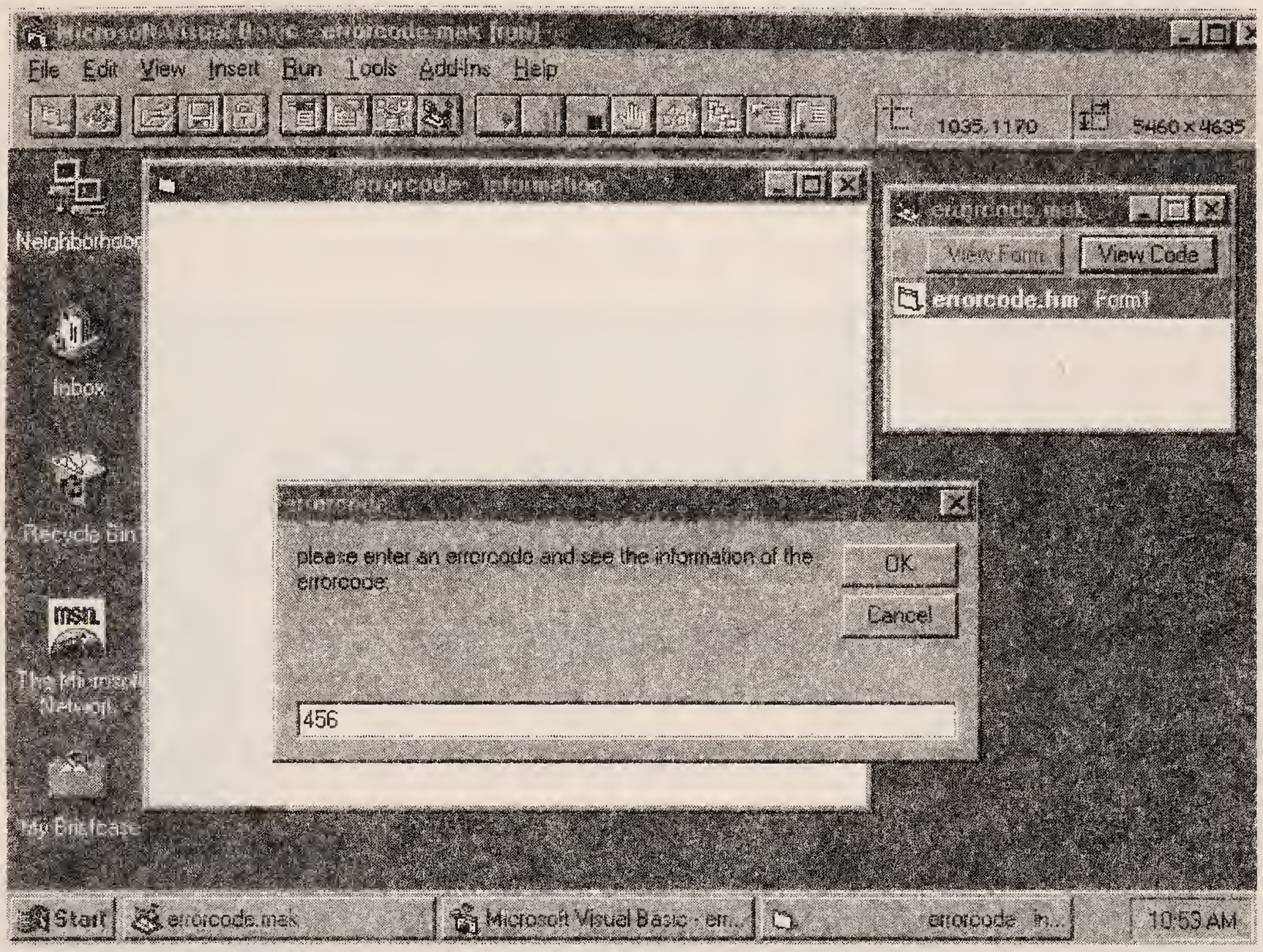
(2) 单击 Project 窗口中的 View Code 按钮，在随即出现的 Code 窗口中选择 Object 列表框中的 Form 以及 Procedure 列表框中的 Click，在 Form _ Click 事件过程头与过程尾之间输入以下代码：

```
Private Sub Form _ Click()
    On Error GoTo ErrorInformation
    Msg="Please enter an errorcode and see the information "
    Msg=Msg & "of the errorcode:"
    errorenter=InputBox(Msg)
    Error ErrorEnter
    Exit Sub
ErrorInformation:
    Msg="the information of the errorcode"
    Msg=Msg & "     " & Err & "     " & "is:" & Chr(10)
    Msg=Msg & "          " & Error(Err) & "          "
    MsgBox(Msg)
    Resume Next
End Sub
```

这段代码的意思是当发生错误时执行 ErrorInformation 错误处理程序。首先，在 Form 窗口中出现一对话框，用户可以输入错误代码。然后，Error ErrorEnter 模拟了一个错误，使程序转而去执行错误处理程序。错误处理程序用以显示一个对话框，显示关于用户输入的错误代码的错误信息。在这里 Chr(10)代表回车。而 Err 函数表示一个标识错误的错误代码。

(3) 关闭 Code 窗口，选择 File 菜单中的 Save Project 命令，在 Save File As 对话框中的 FileName 框中输入文件名 errorcode.frm，单击 OK 按钮。在随后出现的 Save Project As 对话框中的 File Name 框中输入 errorcode.mak，单击 OK 按钮。即将 errorcode 工程文件保存起来了。

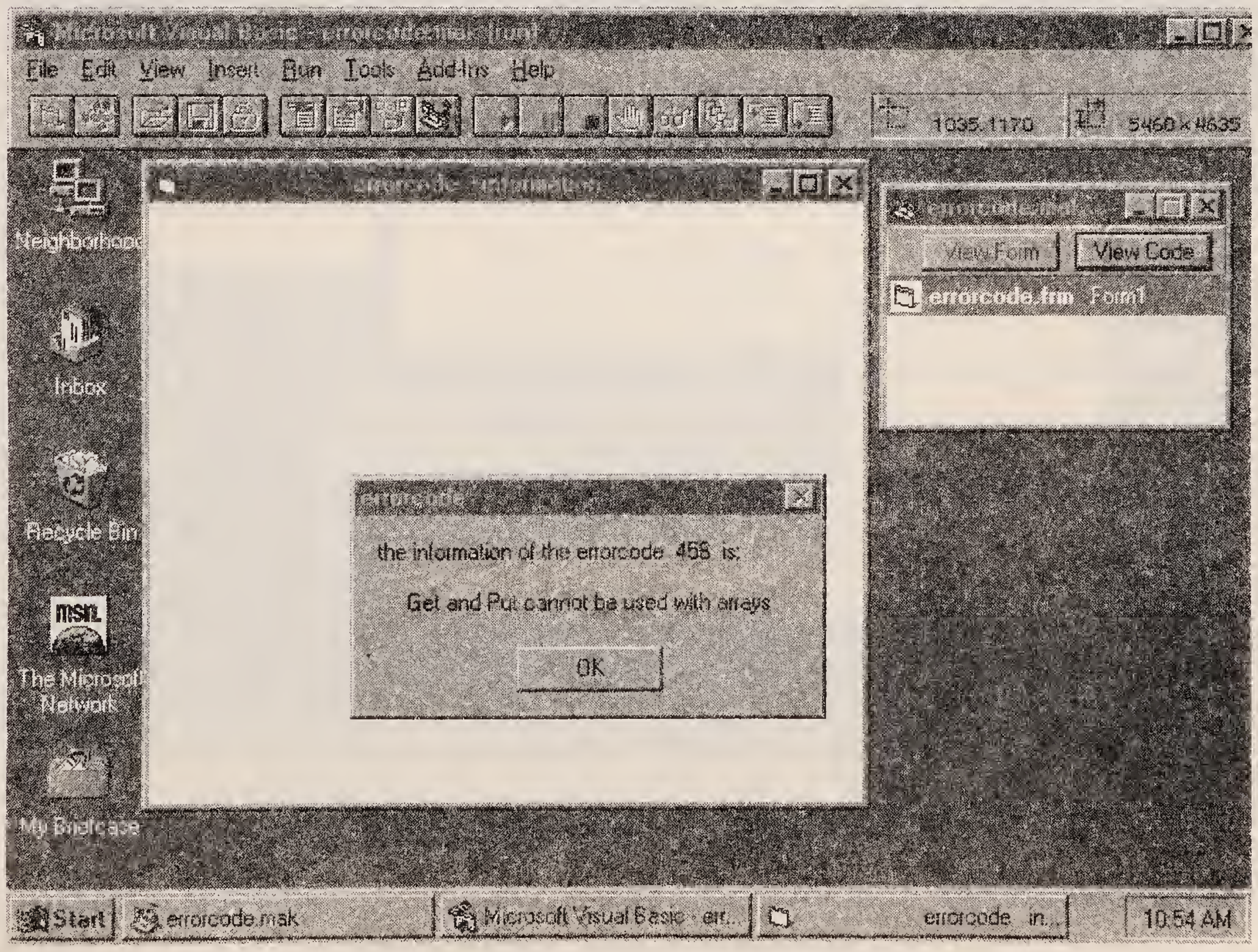
(4) 单击工具栏中的 Start 按钮运行程序。单击 Form 窗口，随即出现一对话框，用户可以随意输入一错误代码，如输入456，然后单击 OK 按钮。如图11.12所示。这时这一对话框消失，接着又出现一对话框，显示错误代码456所代表的错误信息，如图11.13所示。


Microsoft Visual Basic - errorcode.mak [run]
File Edit View Insert Run Tools Add-Ins Help
1035,1170
5460 x 4635
errorcode - information
errorcode.mak
View Form
View Code
errorcode.frm Form1
Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
errorcode
OK
Cancel
Start
errorcode.mak
Microsoft Visual Basic - err...
errorcode - in...
10:53 AM

please enter an errorcode and see the information of the errorcode:

456

图11.12　用户可以在此对话框中输入错误代码


Microsoft Visual Basic - errorcode.mak [run]
File Edit View Insert Run Tools Add-Ins Help
1035,1170
5460 x 4635
errorcode - information
errorcode.mak
View Form
View Code
errorcode.frm Form1
Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
errorcode
OK
Start
errorcode.mak
Microsoft Visual Basic - err...
errorcode - in...
10:54 AM

the information of the errorcode 456 is:

Get and Put cannot be used with arrays

图11.13　此对话框显示与输入错误代码相对应的错误信息

（5）用户可以通过单击 Form 窗口来多次运行此程序，以了解有关错误代码相应的错误信息。

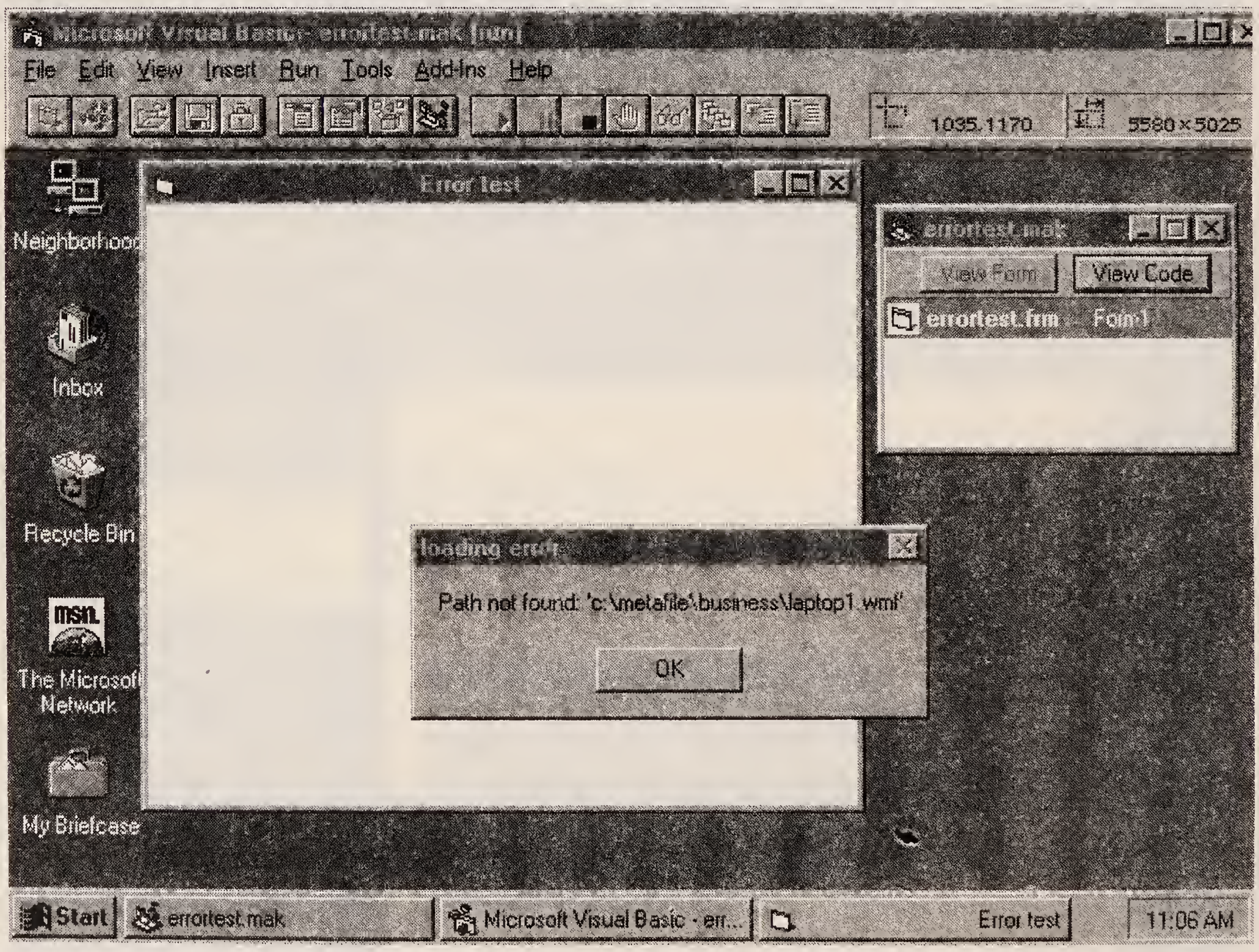
11.2.6　使用 Err.Description 属性

Err 对象拥有一个特殊的属性，可以用在错误处理程序中显示更多的错误信息，这就是 Err.Description 属性。Err.Description 属性在程序运行发生错误时将返回错误信息。用户可以利用这一属性获得更多的错误信息。下面，将前面的 ErrorTest 应用程序稍做改动，将对话框中显示的信息改为由 Err.Description 属性所返回的错误信息。

单击 File 菜单中的 Open Project 命令，在对话框中输入要打开的文件 ERRORTEST.MAK。单击 Project 窗口中的 ViewCode 按钮，在 Code 窗口中对 FileError 错误处理程序作如下修改：

```
FileError:
    MsgBox(Err.Description),,"Loading Error"
    Exit Sub
```

关闭 Code 窗口，单击工具栏中的 Start 按钮运行程序。单击 Form 窗口，随即出现一对话框，显示出错信息“Path not found: ’C:\metafile\business\laptop1.wmf’”。如图11.14所示。这一属性与 Error 的作用是一致的。按错误信息所提示的内容对程序代码进行修改就可以了。


Microsoft Visual Basic - errortest.mak [run]
File Edit View Insert Run Tools Add-Ins Help
1035,1170
5580×5025
Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Error test
errortest.mak
View Form
View Code
errortest.frm
Form1
loading error
OK
Start
errortest.mak
Microsoft Visual Basic - err...
Error test
11:06 AM

Path not found: 'c:\metafile\business\laptop1.wmf'

图11.14　由 Err.Description 属性所提示的错误信息

11.3　使用Debug(调试)窗口

用户在运行程序时所遇到的错误不仅仅是一些语法错误，有时还会经常碰到运行错误或是逻辑错误。除了使用错误处理程序对其进行检测外，用户还可以通过逐行执行代码以及观察变量和属性值来检测错误。要做到这一点，就要在程序运行时进入断点模式，然后在Debug窗口中观察代码的执行情况。进入Debug窗口，可以单击View菜单中的Debug Window，也可以在运行程序时单击Break按钮(在工具栏中)。Debug窗口如图11.15所示。

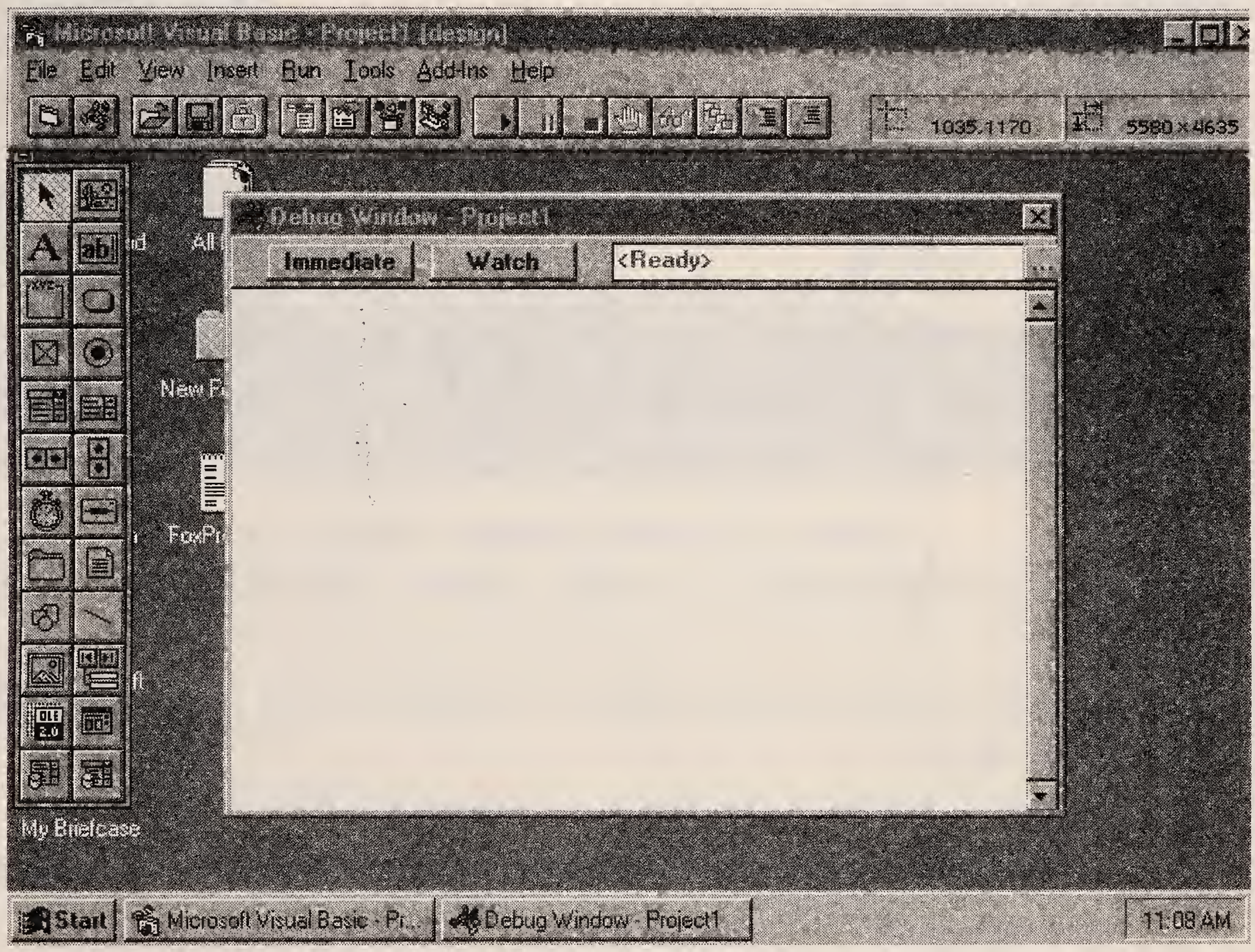

Microsoft Visual Basic - Project1 [design]
File
Edit
View
Insert
Run
Tools
Add-Ins
Help
1035,1170
5580 x 4635
Debug Window - Project1
Immediate
Watch
<Ready>
My Briefcase
Start
Microsoft Visual Basic - Pr...
Debug Window - Project1
11:08 AM

图11.15　Debug窗口

Debug窗口由两部分组成，即观察面板(Watch Pane)和立即面板(Immediate Pane)。这两部分是交替出现的，当开始进入Debug窗口时，窗口内显示的是立即面板，如图11.15所示。在这一面板中允许用户输入并立即执行语句。如想进入观察面板，只要单击Debug窗口上的Watch按钮即可。如图11.16所示。用户可以在Watch面板中对变量值进行观察。Debug窗口为用户提供了一个对程序执行进行观察的有力的工具。

下面我们就举个例子，看看怎样使用Debug窗口来检查和改正错误。这一例子将在Form窗口中画两个文本框，在第一个文本框中用以输入学生的成绩，在第二个文本框中即显示成绩为“Excellent”、“Good”或者是“Pass“、“Not Pass”。

(1) 单击File菜单中的New Project命令，将Form的Caption属性设置为Student Test。

(2) 单击工具框中的文本框工具，在Form窗口中绘制一文本框控件，将其Name属性设置为缺省值text1。Text属性设置为空，即没有任何字符。

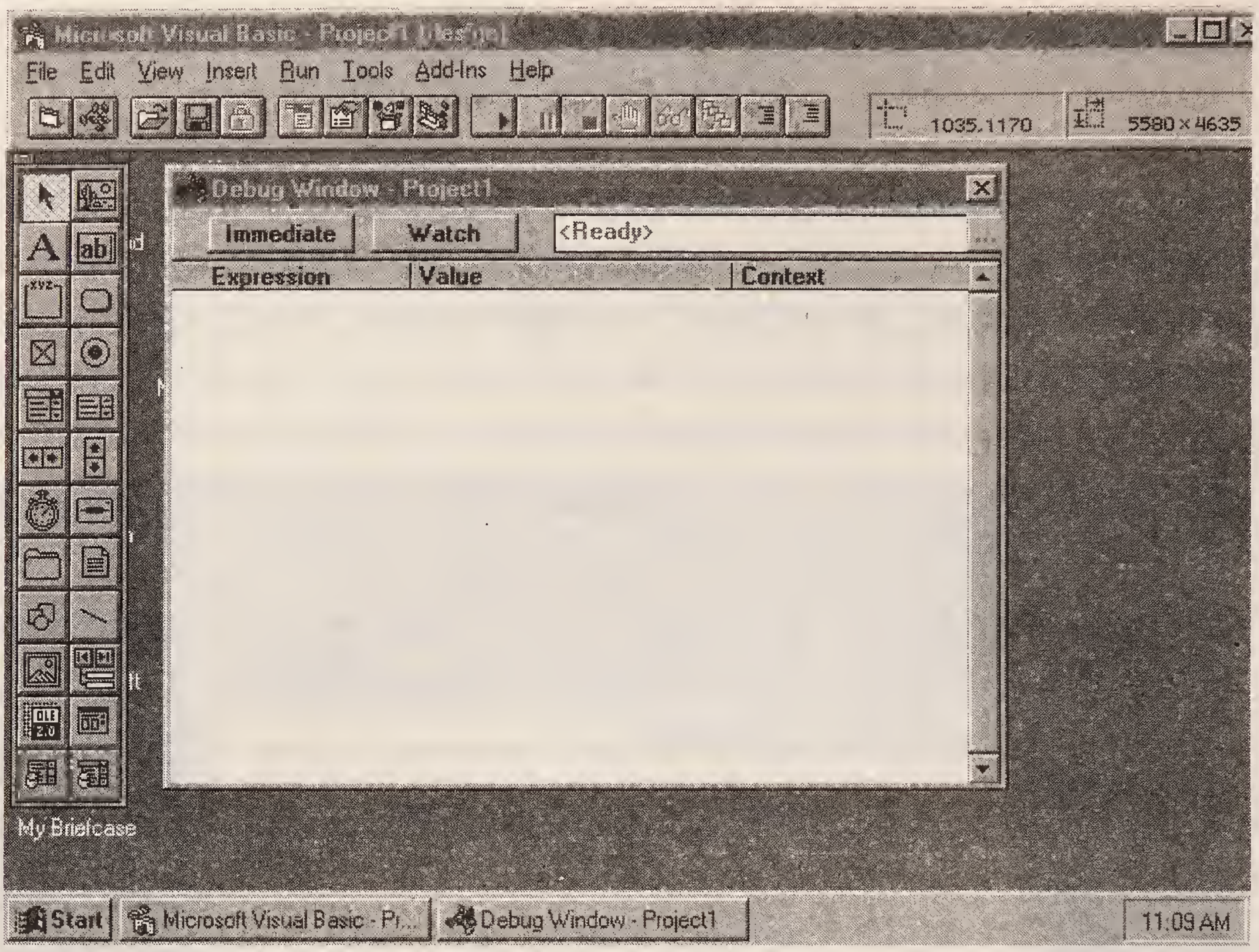

Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
1035,1170
5580×4635
Debug Window - Project1
Immediate
Watch
<Ready>
Expression
Value
Context
My Briefcase
Start
Microsoft Visual Basic - Pr...
Debug Window - Project1
11:09 AM

图11.16　Debug 窗口中的 Watch 面板

（3）再单击工具框中的标签按钮，在文本框上方绘制一标签，将其 Caption 属性设置为 Student’s Score。

（4）再使用文本框工具在 Form 窗口中绘制一文本框控件，将其 Name 属性设置为缺省值 text2。Text 属性设置为空。

（5）在文本框控件上方绘制一标签，其 Caption 属性为 Degree。

（6）使用工具框中的命令按钮工具在 Form 窗口中绘制一命令按钮，将其 Caption 属性设置为 Erase，Name 属性设为 Clear。再使用命令按钮工具在 Form 窗口中绘制另一命令按钮控件，将其 Caption 和 Name 属性均设置为 test。

这样在 Form 窗口中绘制控件的工作就完成，其 Form 窗口如图11.17所示。

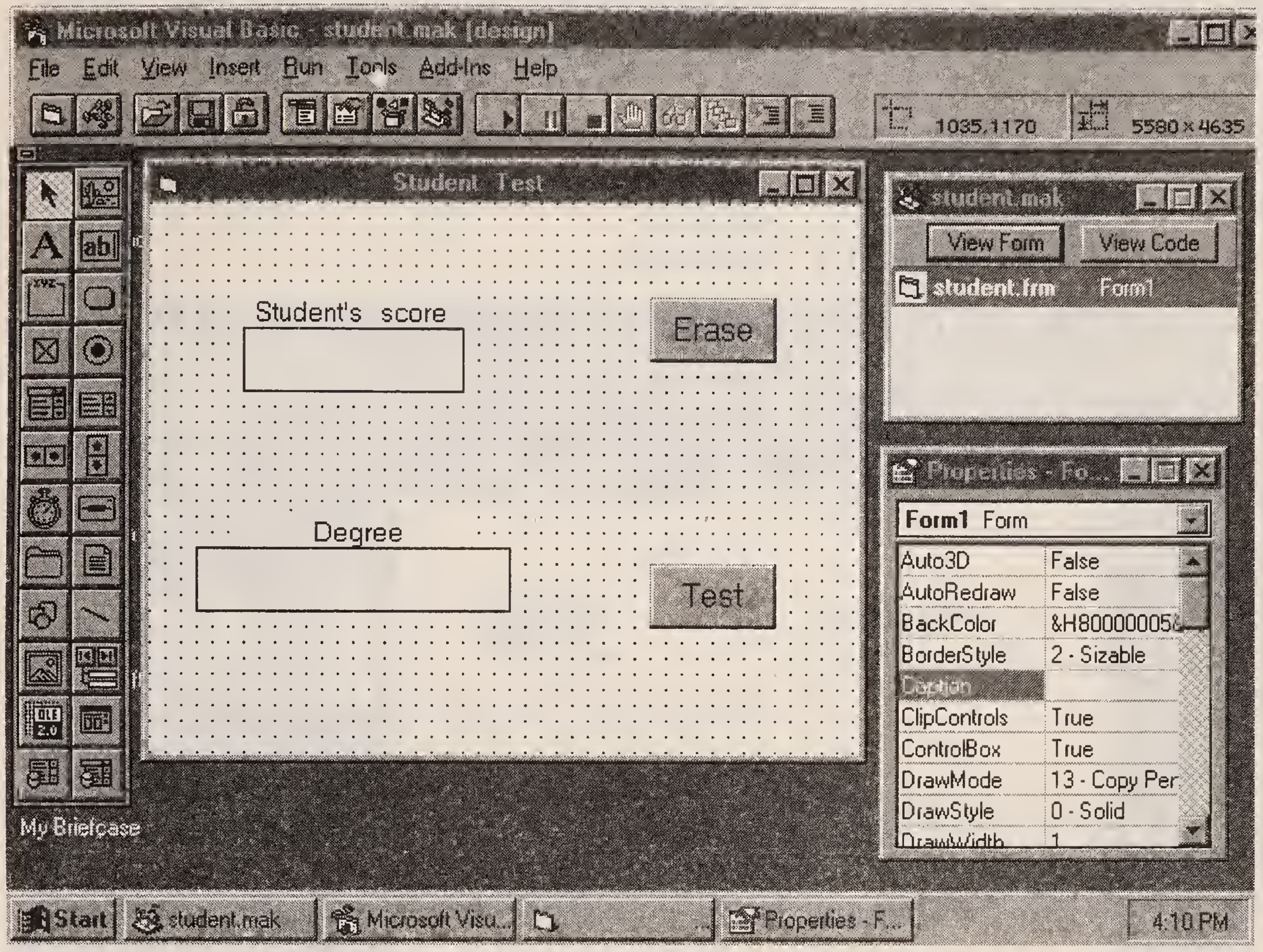
（7）现在就要为控件进行连接代码的工作了。单击 Project 窗口中的 View Code 按钮，或双击 Form 窗口，即可进入代码窗口。选择 Object 列表框中的 Clear 和 Procedure 列表框中的 Click，在 Clear _ Click()事件过程头与过程尾之间输入下列代码：

```
Private Sub Clear _ Click()
    text1. Text=""
    text2. Text=""
End Sub
```

这段代码的意思是清除 text1与 text2文本框中的内容。

（8）选择 Object 列表框中的 test 与 Procedure 列表框中的 Click，在 test _ Click()事件的过程头与过程尾之间输入下列代码：

```
Private Sub test _ Click()
    Dim x As Single
    x=Val(text1. Text)
```


Microsoft Visual Basic - student.mak [design]
File Edit View Insert Run Tools Add-Ins Help
1035,1170
5580 x 4635
Student Test
Student's score
Erase
Degree
Test
student.mak
View Form
View Code
student.frm Form1
Properties - Fo...
Form1 Form
My Briefcase
Start
student.mak
Microsoft Visu...
Properties - F...
4:10 PM

Auto3D	False
AutoRedraw	False
BackColor	&H80000005&
BorderStyle	2 - Sizable
Caption	
ClipControls	True
ControlBox	True
DrawMode	13 - Copy Per
DrawStyle	0 - Solid
DrawWidth	1

图11.17 StudentTest 工程文件的 Form 窗口

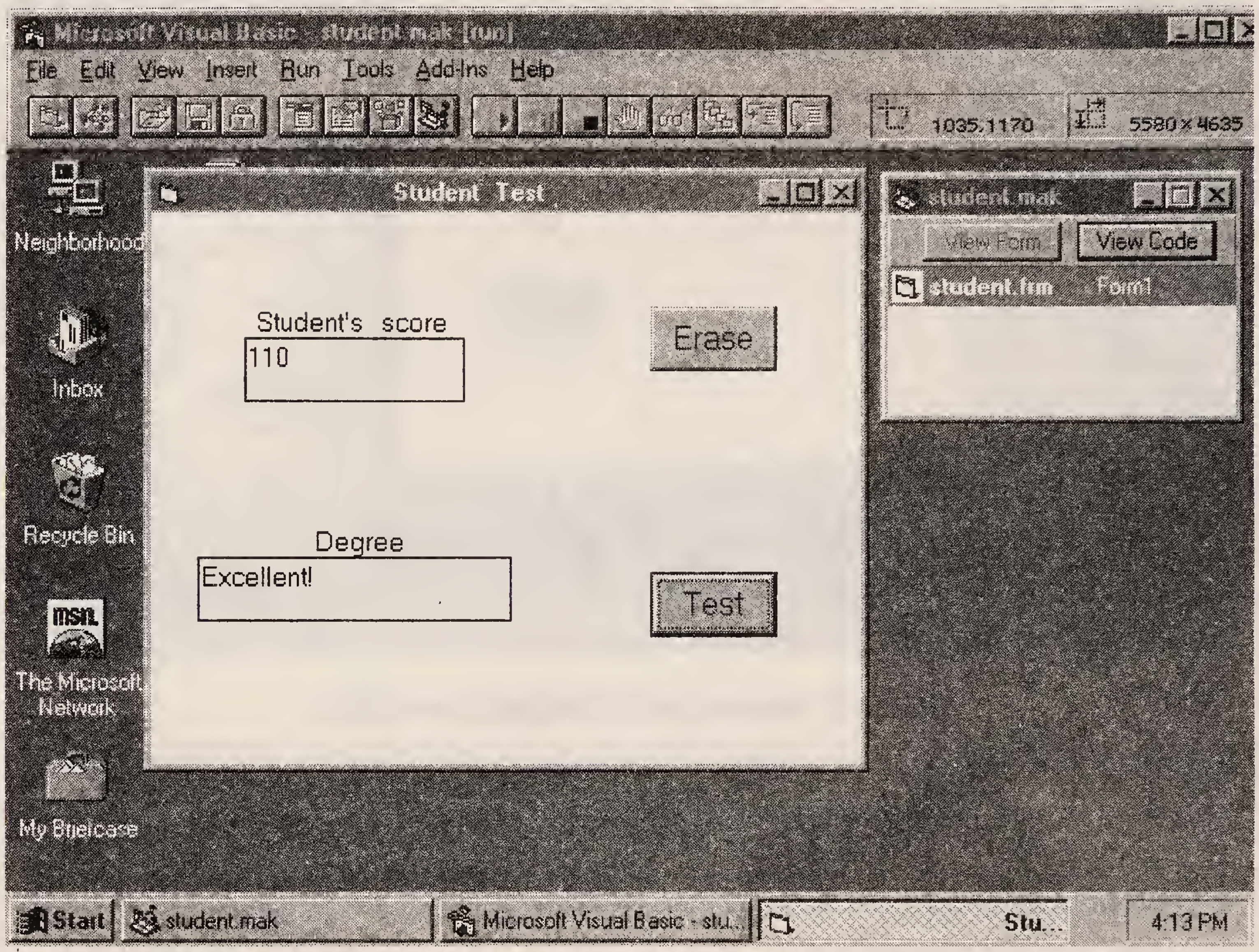
```
    If x > 100 Then
    text2.Text="You enter the wrong score!"
    End If
    If x >= 90 Then
    text2.Text="Excellent!"
    End If
    If x >= 70 And x < 90 Then
    text2.Text="Good!"
    End If
    If x >= 60 and x < 70 Then
    text2.Text="Pass!"
    End If
    If x < 60 Then
    text2.Text="Not Pass!"
    End If
End Sub
```

这段代码的意思是当输入的分数大于100时，text2文本框输出“You enter the wrong score!”的内容。如果输入的分数介于90到100之间时，输出对学生的评语为 Excellent（优秀）。当分数介于70到90之间时，评语为 Good（良好）。当分数大于等于60分而小于70分时，评语为 Pass（通过）。当分数于小60分时，则表示 Not Pass（没通过）。

（9）关闭 Code 窗口，选择 File 菜单中的 Save Project 命令，在随后出现的 Save File As 对话框中在 C:\VB 路径下输入文件名为 STUDENT.FRM。单击 OK 按钮，在随后出现的 Save Project 对话框中在同一路径下输入文件名 STUDENT.MAK。单击 OK 按钮，将文件

保存起来。

(10) 单击工具栏中的 Start 按钮，运行程序。在运行程序时发现，当我们往 text1 文本框（即运行程序时就出现光标的文本框）中输入大于100的数，如110时，单击 Test 按钮，在第二个文本框中并没有输出当初预想的“You enter the wrong score!”语句，而是输出了“Excellent!”。如图11.18所示，这说明发生了语法错误。


Microsoft Visual Basic - student.mak [run]
File Edit View Insert Run Tools Add-Ins Help
1035,1170
5580 x 4635
Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Student Test
Erase
Test
student.mak
View Form
View Code
student.frm Form1
Start
student.mak
Microsoft Visual Basic - stu...
Stu...
4:13 PM

Student's score
110

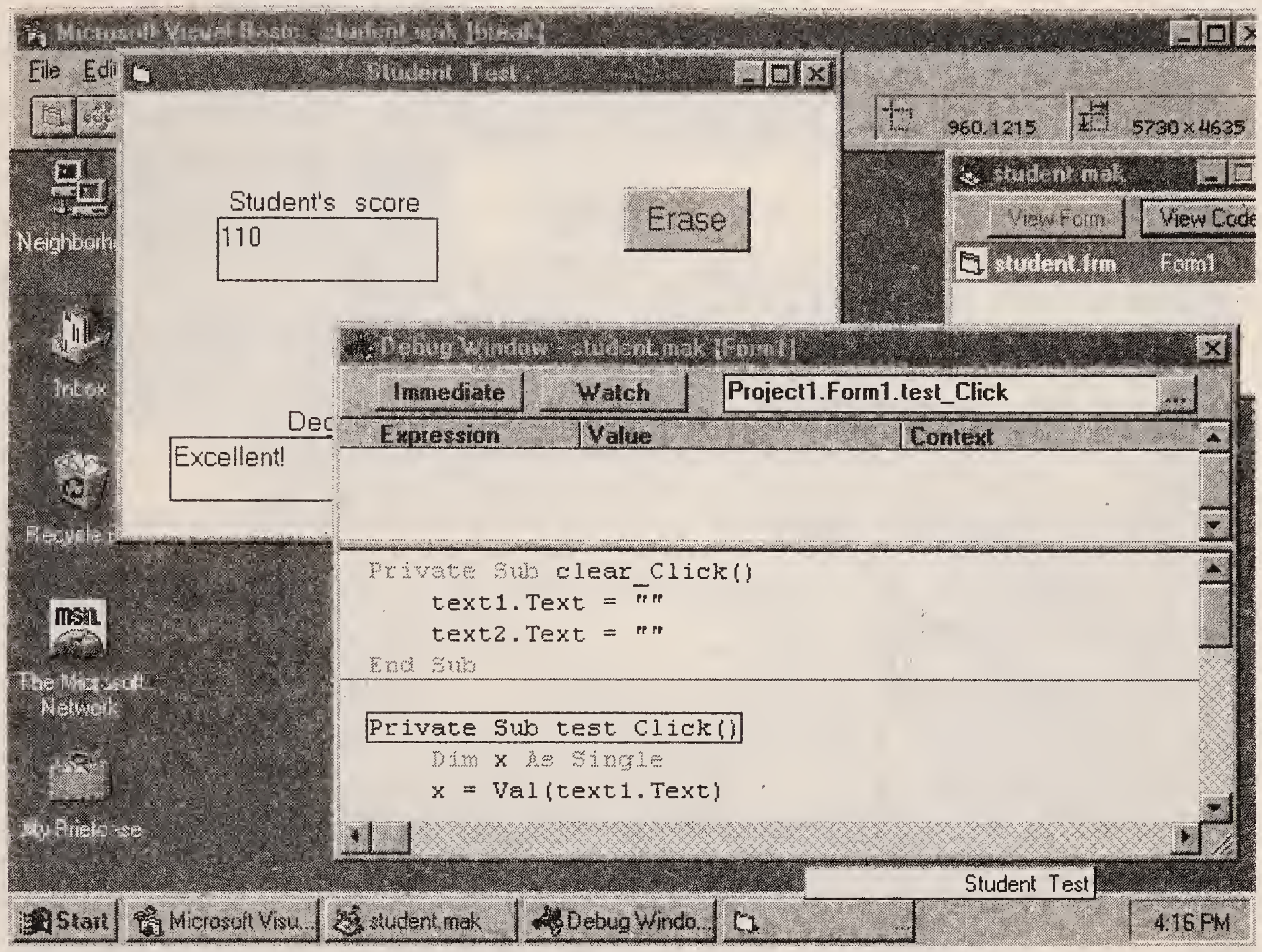
Degree
Excellent!

图11.18　程序中发生逻辑错误引起的输出错误

(11) 单击菜单栏中的 Break 按钮（在 Start 与 End 按钮之间），暂停程序的运行。这时 Visual Basic 将显示出 Debug 窗口。这时的 Debug 窗口上半部分是空的，并且当前面板是 Immediate（立即）面板。如想转入 Watch（观察）面板，单击 Debug 窗口上的 Watch 按钮即可。Debug 窗口的下半部分是程序代码。

(12) 单击工具栏上的 Step Into 按钮（在工具栏右边数倒数第二个按钮即是。当鼠标指针移到按钮图标上时，会在其下方出现一小方框，内写 Step Into），执行程序的下一语句。这时 Visual Basic 又将控制权交还给程序。

(13) 单击 Test 按钮，这时在 Debug 窗口下半部分的程序代码第一句 Private Sub test _ Click()被一细线框体包围起来，如图11.19所示。用鼠标选择 X，将其用高亮覆盖，然后单击工具栏中的 Instant Watch 按钮（带有一幅眼镜的图标），即出现 Instant Watch 对话框，如图11.20所示，对话框中顶部的“Current Context:Project1.Form1.test _ Click”表示选中的表达式或变量 X 所在的过程、窗体以及工程文件。Expression 表示选中的表达式或变量，Value 则为表达式或变量的值。单击 Add Watch 按钮，这样，就可以在 Debug 窗口中的观察面板中观察变量 X 的值了。


Microsoft Visual Basic - student.mak [break]
File
Edit
Student Test
960,1215
5730 x 4635
student.mak
View Form
View Code
student.frm
Form1
Student's score
Erase
Debug Window - student.mak [Form1]
Immediate
Watch
Expression
Value
Context
Student Test
Start
Microsoft Visu...
student.mak
Debug Windo...
4:16 PM

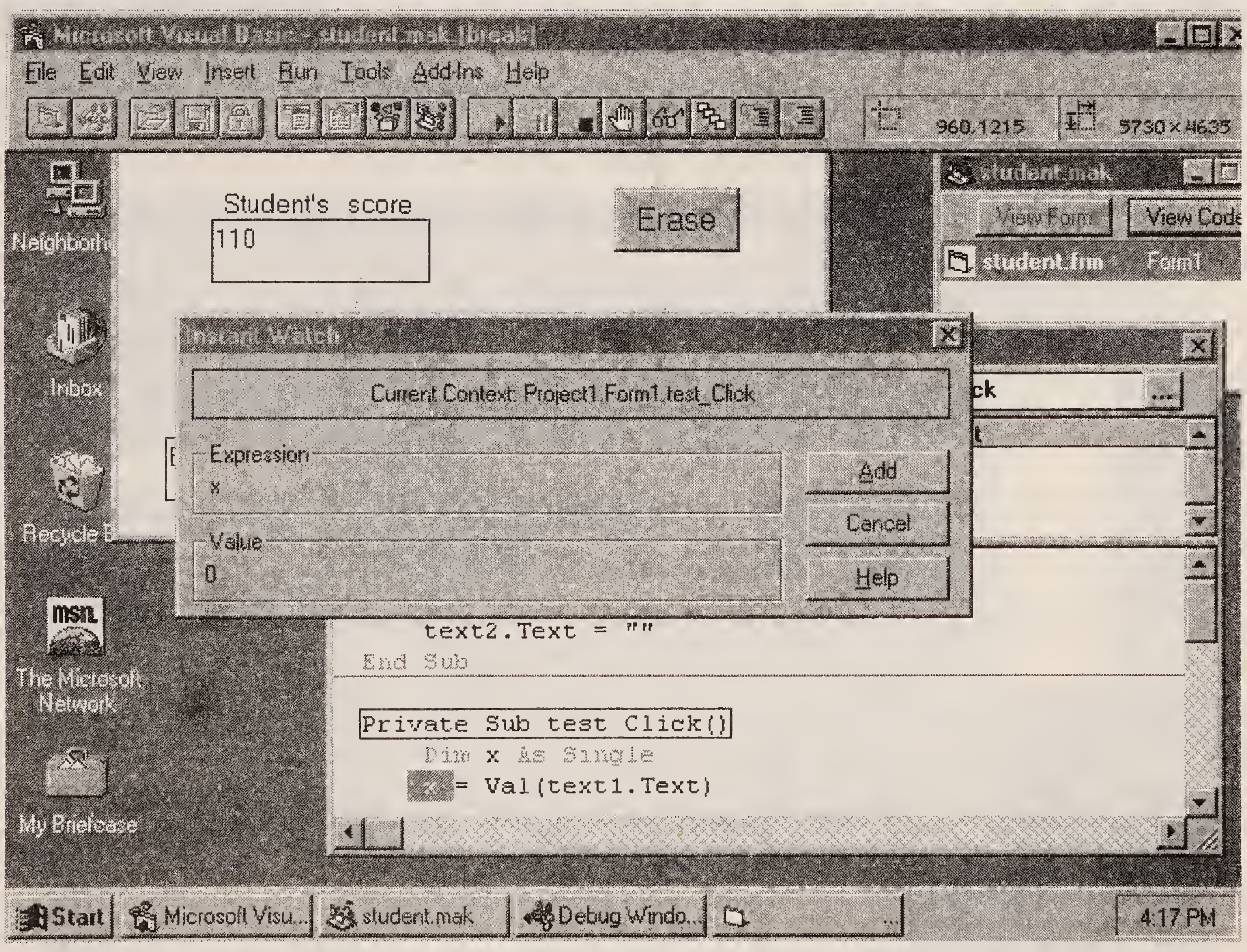
Student's score: 110

Excellent!

Project1.Form1.test_Click

```
Private Sub clear_Click()
    text1.Text = ""
    text2.Text = ""
End Sub

Private Sub test_Click()
    Dim x As Single
    x = Val(text1.Text)
```

图11. 19　单步执行时的程序代码


Microsoft Visual Basic - student.mak [break]
File Edit View Insert Run Tools Add-Ins Help
960,1215
5730 x 4635
student.mak
View Form
View Code
student.frm
Form1
Student's score
Erase
Instant Watch
Add
Cancel
Help
Start
Microsoft Visu...
student.mak
Debug Windo...
4:17 PM

Student's score: 110

Current Context: Project1.Form1.test_Click

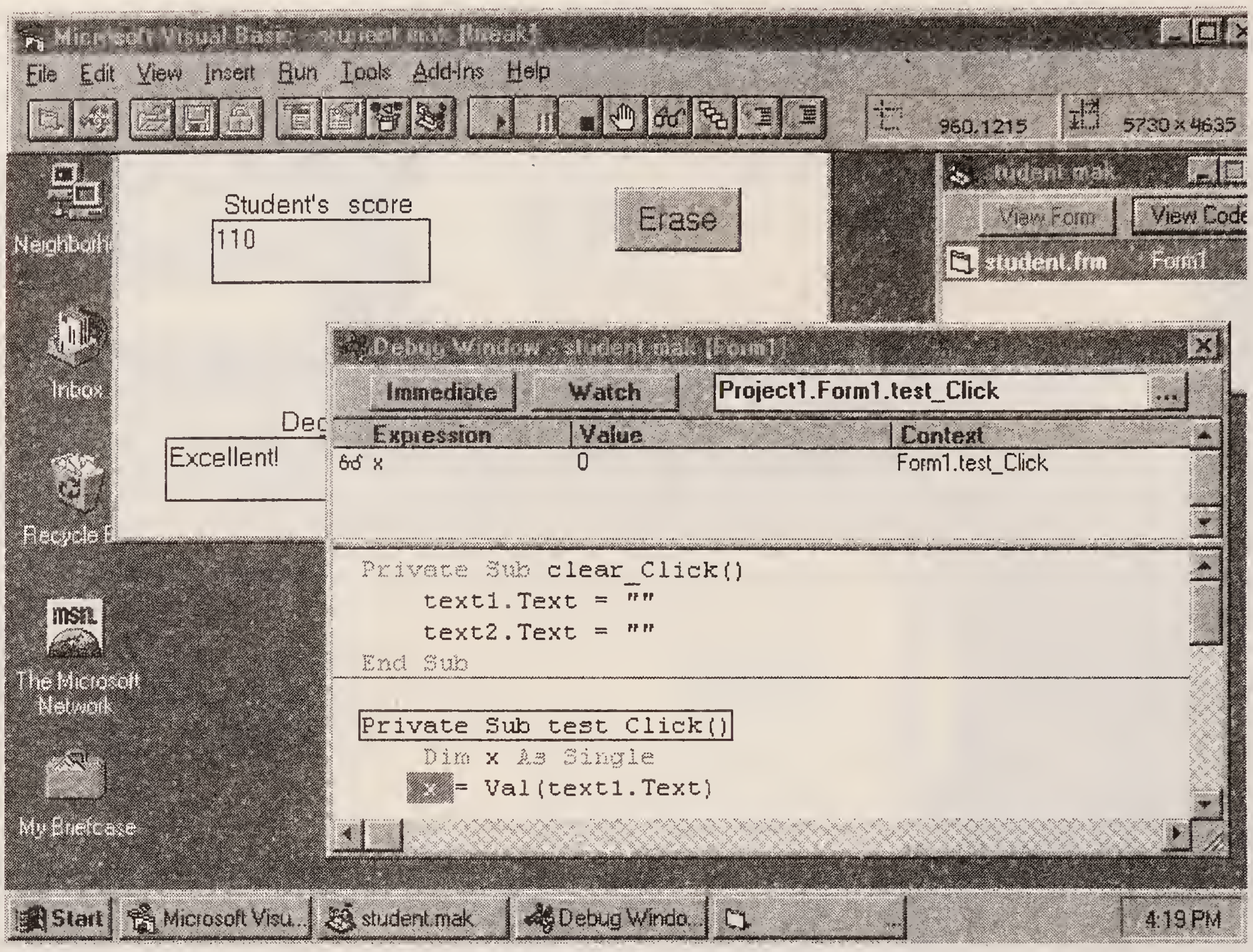
Expression: x

Value: 0

```
    text2.Text = ""
End Sub

Private Sub test_Click()
    Dim x As Single
    x = Val(text1.Text)
```

图11. 20　Instant Watch(立即观察)对话框

(14) 单击 Debug 窗口中 Watch 按钮,即在窗口的上半部分出现刚才选中的观察变量,以及它的变量值和所在的窗体及过程,如图11. 21所示。由于这时还没有为 X 赋值,所以它的变量值为0。


Microsoft Visual Basic - student.mak [break]
File Edit View Insert Run Tools Add-Ins Help
960,1215
5730 x 4635
Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Erase
student.mak
View Form
View Code
student.frm
Form1
Debug Window - student.mak [Form1]
Immediate
Watch
Start
Microsoft Visu...
student.mak
Debug Windo...
4:19 PM

Student's score: 110

Excellent!

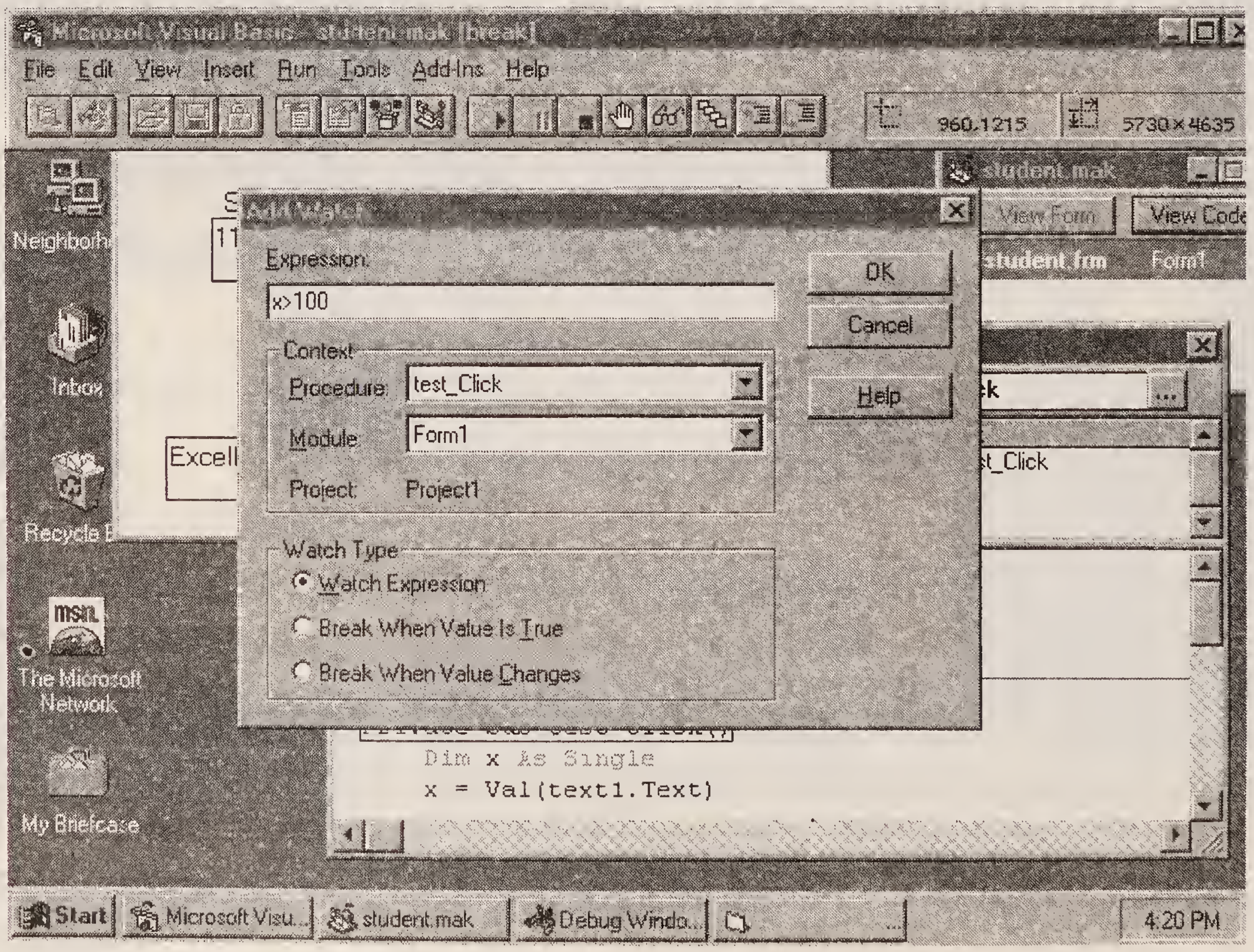
Project1.Form1.test_Click

Expression	Value	Context
x	0	Form1.test_Click

```
Private Sub clear_Click()
    text1.Text = ""
    text2.Text = ""
End Sub

Private Sub test_Click()
    Dim x As Single
    x = Val(text1.Text)
```

图11.21　Watch 面板中的变量及变量值

（15）除了使用 Instant Watch 按钮在 Wathc 面板中加入观察变量外，用户还可以使用 Tools 菜单中的 Add Watch 命令来添加观察变量。单击 Tools 菜单中的 Add Watch 命令，即出现一对话框，如图11.22所示。在对话框中键入 x>100，然后单击 OK 按钮，即在 Debug 窗口的 Watch 面板中出现 x>100表达式以及它的当前值 False。


Microsoft Visual Basic - student.mak [break]
File Edit View Insert Run Tools Add-Ins Help
960,1215
5730 x 4635
Add Watch
OK
Cancel
Help
student.mak
View Form
View Code
student.frm
Form1
Start
Microsoft Visu...
student.mak
Debug Windo...
4:20 PM

Expression: x>100

Context
Procedure: test_Click
Module: Form1
Project: Project1

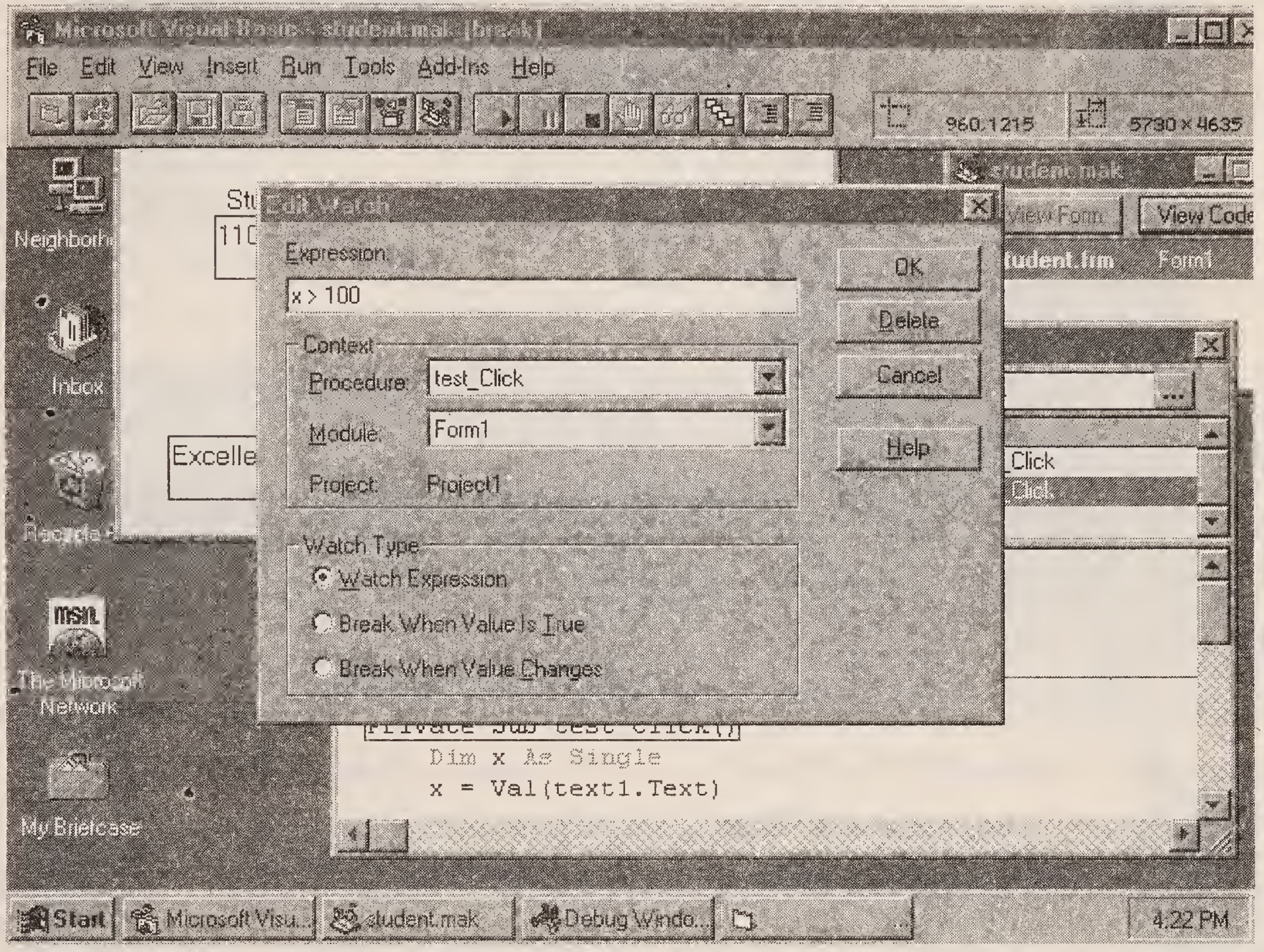
Watch Type
● Watch Expression
○ Break When Value Is True
○ Break When Value Changes

```
    Dim x As Single
    x = Val(text1.Text)
```

图11.22　Add Watch（加入观察变量）对话框

(16) 使用 Instant Watch 按钮，在 Debug 窗口中的 Watch 面板上加入 x>=90表达式，来观察它的变量值，这时它的值为 False。因为 x 还未赋值。

(17) 如果用户想要删掉某一观察变量，双击 Watch 面板中这一变量或表达式，即在屏幕上弹出 Edit Watch 对话框，单击 Delete 按钮即可。如图11.23所示。


Microsoft Visual Basic - student.mak [break]
File Edit View Insert Run Tools Add-Ins Help
960,1215
5730 x 4635
student.mak
View Form
View Code
student.frm
Form1
Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
110
Excelle
Edit Watch
OK
Delete
Cancel
Help
Click
Click
Start
Microsoft Visu...
student.mak
Debug Windo...
4:22 PM

Expression: x > 100

Context
Procedure: test_Click
Module: Form1
Project: Project1

Watch Type
● Watch Expression
○ Break When Value Is True
○ Break When Value Changes

```
Private Sub test_Click()
    Dim x As Single
    x = Val(text1.Text)
```

图11.23　Edit Watch（编辑观察变量）对话框

(18) 在 Watch 面板中加入 x 与 x>100及 x>=90表达式以后，单击 Step Into 按钮，执行"Private Sub test _ Click()"语句。同时"x=Val(text1.text)"语句被一框体包围起来。

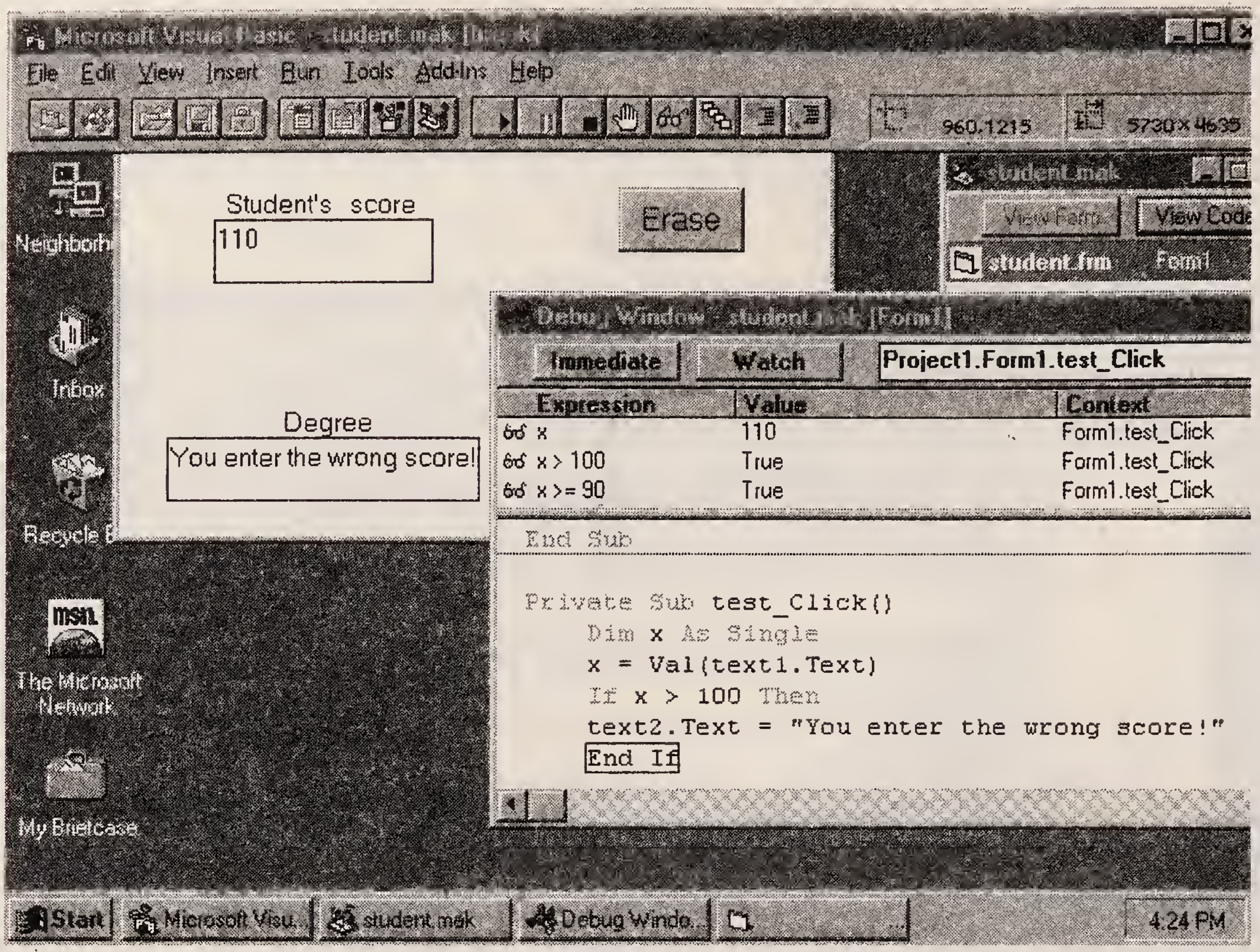
(19) 单击 Step Into 按钮，执行刚才细线框内的语句，即把 text1文本框内的值赋给 x，这时 Watch 面板变量 x 的值变为110。

(20) 继续单击 Step Into 按钮，当其执行"text2.Text="You enter the wrong score!"语句时，在 text2文本框中输出了"You enter the wrong score!"，同时 x>100以及 x>=90表达式的 Value 值都为 True。这说明程序在运行时执行过这一语句。如图11.24所示。那为什么我们没有看到呢?

(21) 继续单击 Step Into 按钮，当执行"Excellent!"语句时，在 Text2文本框中输出了"Excellent!"内容。继续单击 Step Into 按钮，直到这一过程结束。我们发现 Text2文本框中的内容一直保持为"Excellent"，并且 x>100以及 x>=90的 Value 值也一直为 True。

(22) 这样，就找到出错原因了。原来当输入值 x 为110，当执行 x>100语句时，在 text2文本框中输出了正确的结果"You enter the wrong score!"。当程序继续运行，执行 x>=90语句时，又在 text2文本框中输出了错误的结果"Excellent!"，从而覆盖了原来的内容。因而最后我们只看到 text2文本框中的"Excellent!"内容。

(23) 既然找到了错误，就要对其进行修改。用户可以结束程序运行，回到 Code 窗口对代码进行修改，也可以在 Debug 窗口下半部分的程序代码中直接对其进行修改。在这里注


Microsoft Visual Basic - student.mak [break]
File Edit View Insert Run Tools Add-Ins Help
960,1215
5730 x 4635
student.mak
View Form
View Code
student.frm
Form1
Erase
Debug Window - student.mak [Form1]
Immediate
Watch
Start
Microsoft Visu...
student.mak
Debug Windo...
4:24 PM
Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase

Student's score: 110

Degree: You enter the wrong score!

Project1.Form1.test_Click

Expression	Value	Context
x	110	Form1.test_Click
x > 100	True	Form1.test_Click
x >= 90	True	Form1.test_Click

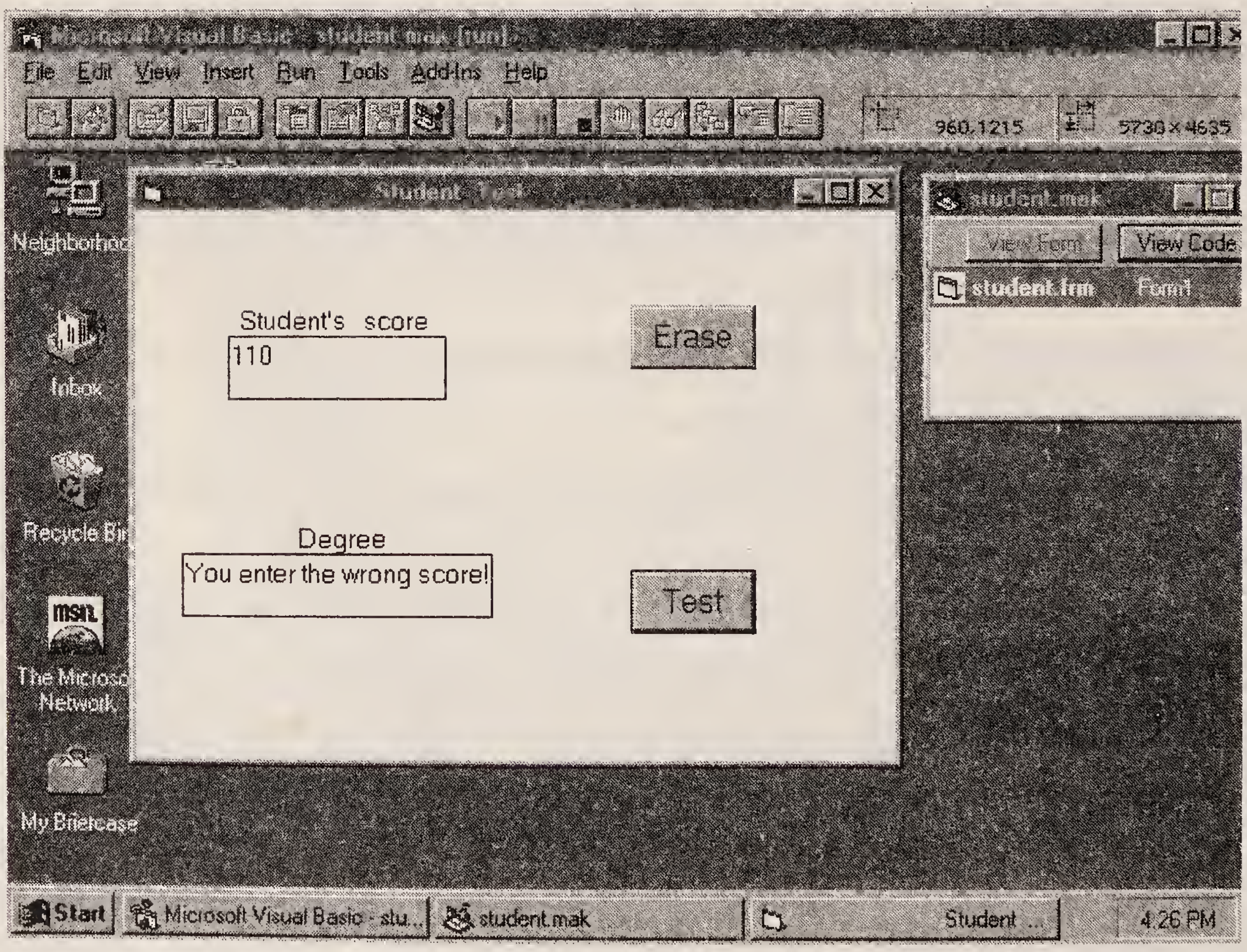
```
End Sub

Private Sub test_Click()
    Dim x As Single
    x = Val(text1.Text)
    If x > 100 Then
    text2.Text = "You enter the wrong score!"
    End If
```

图11.24　单步运行“text2.Text=“You enter the wrong score!”语句

意，在单步执行情况下，如果修改的这一句已被执行过，则直到再次运行前，它不会产生影响。在 Debug 窗口中的程序代码中找到 If x>=90 Then... 语句，将其改为：

If x>=90 And x<=100 Then

（24）单击工具栏中的 Start 按钮，在 text1文本框中输入110，单击 Test 按钮，这时在 text2文本框中输入出正确内容“You enter the wrong score!”，如图11.25所示。


Microsoft Visual Basic - student.mak [run]
File Edit View Insert Run Tools Add-Ins Help
960,1215
5730 x 4635
Student Test
student.mak
View Form
View Code
student.frm
Form1
Erase
Test
Start
Microsoft Visual Basic - stu...
student.mak
Student ...
4:26 PM
Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase

Student's score: 110

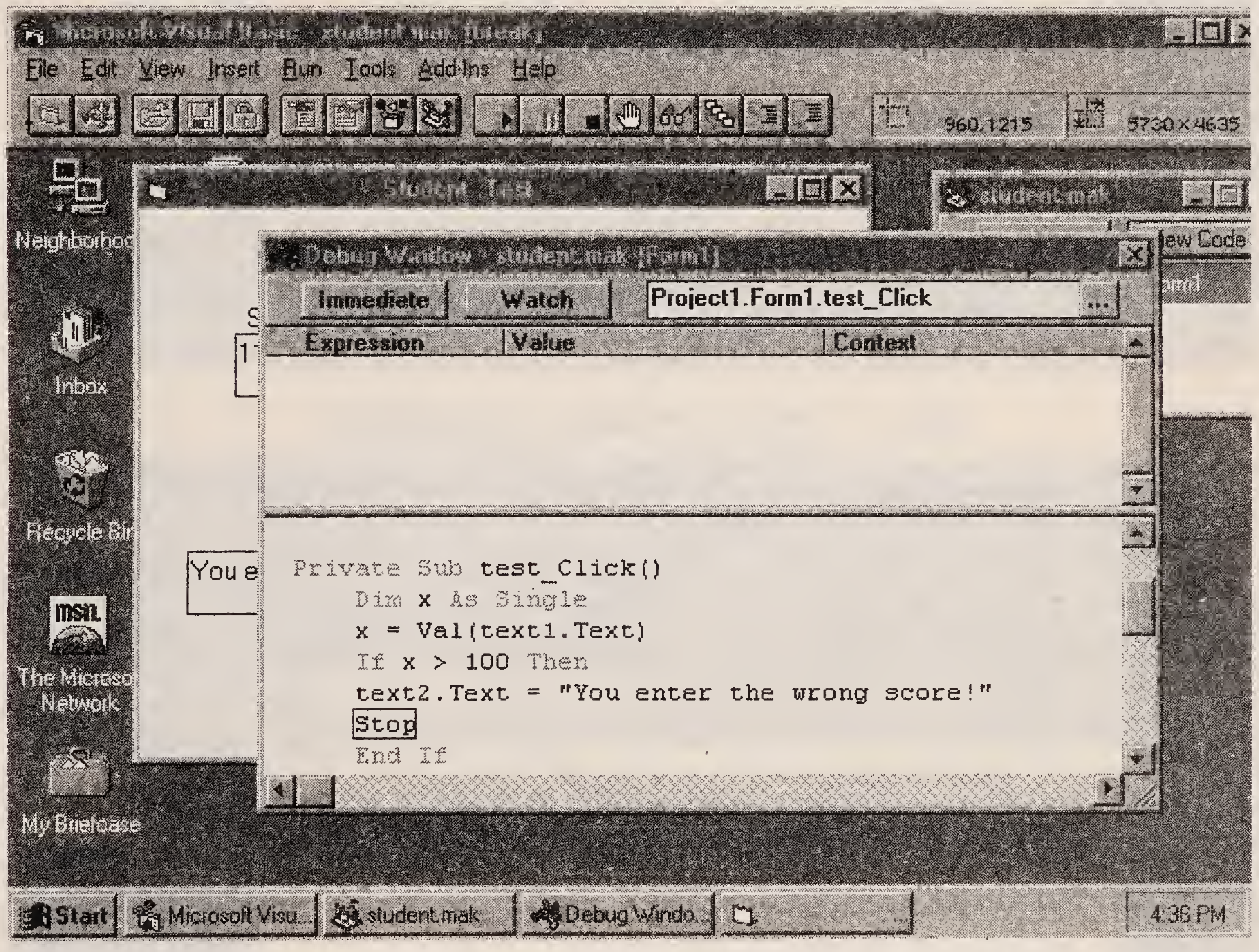
Degree: You enter the wrong score!

图11.25　程序运行结果正确

这样，寻找并修改错误的过程就完成了。

11.4　设置断点

如果用户希望程序在执行到某一段程序时，自动停止并弹出 Debug 窗口，可以通过在程序中添加 Stop 语句来实现。Stop 语句放在用户希望程序停止执行的语句的前面。例如，在 Student 工程文件中，在 text2. Text="You enter the wrong score!"语句后加 Stop 语句，则在运行程序时当 Visual Basic 执行到 Stop 语句时即停止程序的运行。同时弹出 Debug 窗口，在 Debug 窗口下半部分的程序代码中，Stop 语句被一细线框所包围。如图11. 26所示。这时用户可以通过单击 Step Into 按钮来继续执行 Stop 下面的语句。

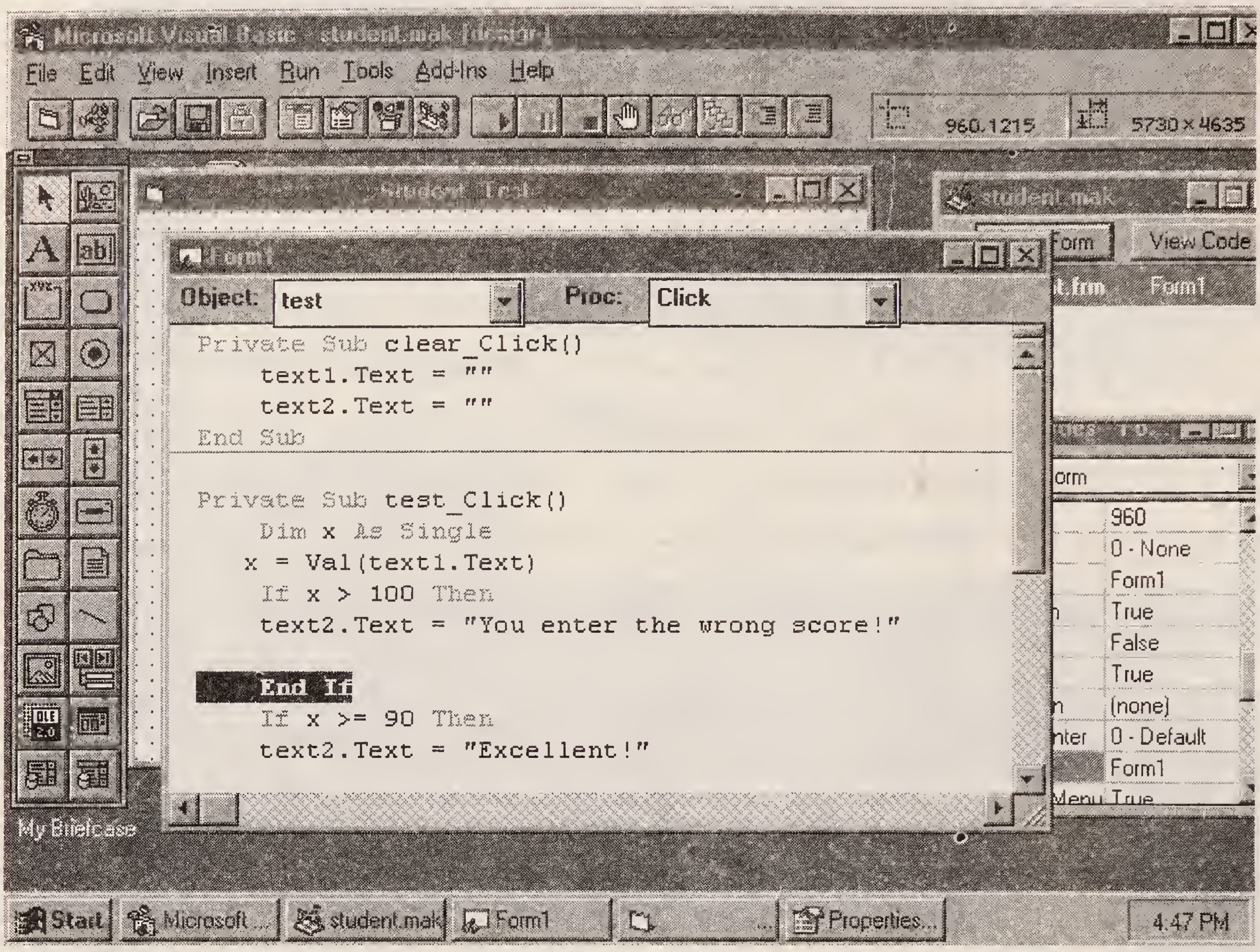

Microsoft Visual Basic - student.mak [break]
File Edit View Insert Run Tools Add-Ins Help
960,1215
5730 x 4635
Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Student Test
student.mak
View Code
Debug Window - student.mak [Form1]
Immediate
Watch
Project1.Form1.test_Click
Expression
Value
Context
You e
Start
Microsoft Visu...
student.mak
Debug Windo...
4:36 PM

```
Private Sub test_Click()
    Dim x As Single
    x = Val(text1.Text)
    If x > 100 Then
    text2.Text = "You enter the wrong score!"
    Stop
    End If
```

图11. 26　执行 Stop 语句

用户还可以通过使用工具栏中的 Toggle Breakpoint(设置断点)按钮来设置断点。首先将光标移到想要设置断点的那行语句，然后单击 Toggle Breakpoint 按钮，则这一行将被红色高亮所覆盖。如图11. 27所示。当运行程序，Visual Basic 执行这一语句时即暂停程序的运行且弹出 Debug 窗口，并且将设置断点的那行语句用一细线框包围起来，并且在语句前有一红色高亮，如图11. 28所示。

想要删除设置的断点，将光标移到想要删除断点的语句前面，然后单击工具栏中的 Toggle Breakpoint 按钮即可。如果设置断点过多，可以选择 Run 菜单中的 Clear All Breakpoints 命令。


Microsoft Visual Basic - student.mak [design]
File Edit View Insert Run Tools Add-Ins Help
960,1215
5730 x 4635
Student Test
student.mak
Form
View Code
t.frm
Form1
My Briefcase
Start
Microsoft ...
student.mak
Form1
Properties...
4:47 PM

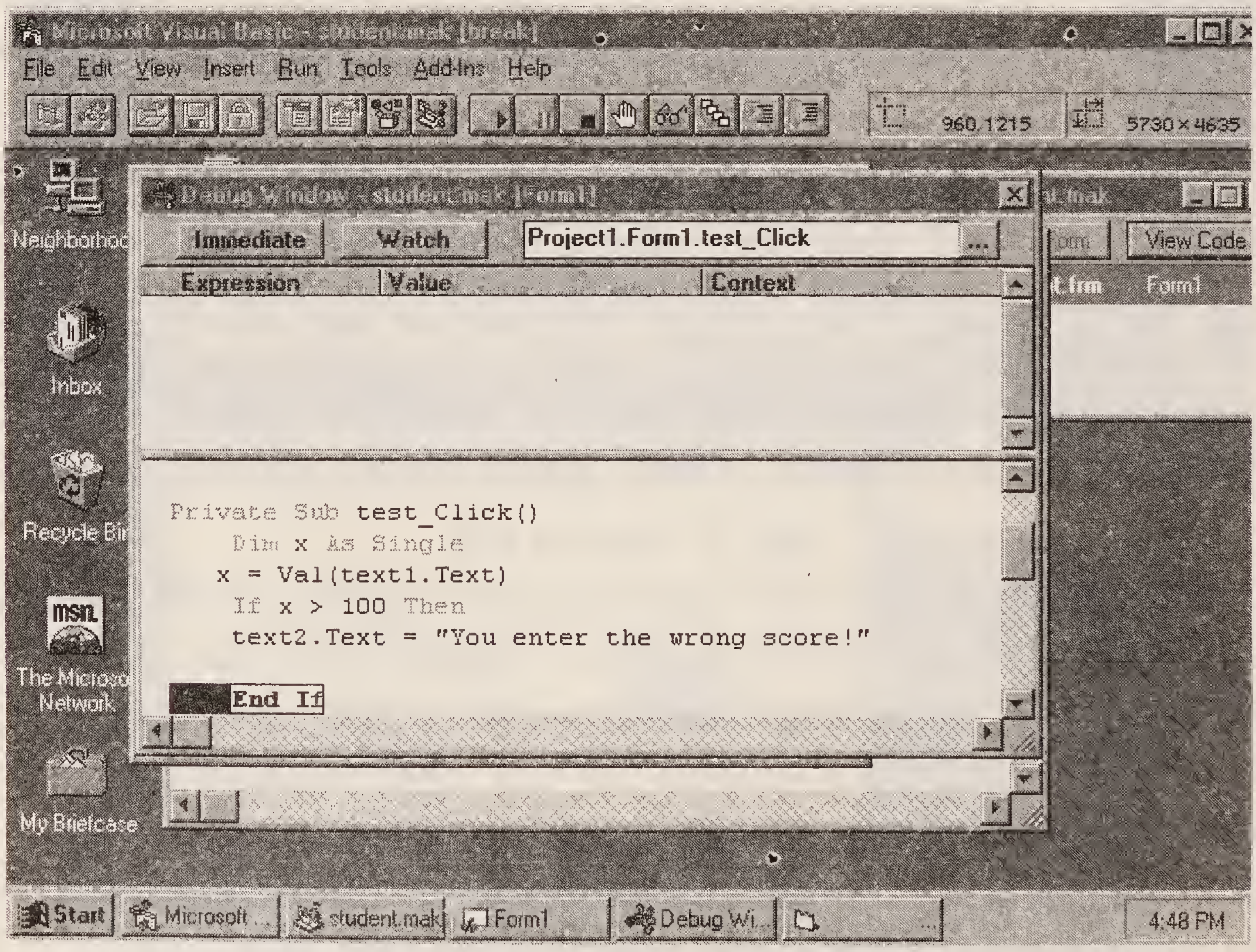
Form1

Object: test Proc: Click

```
Private Sub clear_Click()
    text1.Text = ""
    text2.Text = ""
End Sub

Private Sub test_Click()
    Dim x As Single
   x = Val(text1.Text)
    If x > 100 Then
    text2.Text = "You enter the wrong score!"

    End If
    If x >= 90 Then
    text2.Text = "Excellent!"
```

960
0 - None
Form1
True
False
True
(none)
0 - Default
Form1
True

图11. 27　用 Toggle Breakpoint 按钮设置断点


Microsoft Visual Basic - student.mak [break]
File Edit View Insert Run Tools Add-Ins Help
960,1215
5730 x 4635
Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Immediate
Watch
Expression
Value
Context
View Code
t.frm
Form1
Start
Microsoft ...
student.mak
Form1
Debug Wi...
4:48 PM

Debug Window - student.mak [Form1]

Project1.Form1.test_Click

```
Private Sub test_Click()
    Dim x As Single
   x = Val(text1.Text)
    If x > 100 Then
    text2.Text = "You enter the wrong score!"

    End If
```

图11. 28　程序运行时遇到断点

第十二章　Visual Basic 4.0高级技术

Microsoft Windows 95是一个多任务操作系统，可以运行多个应用程序，用户将在本章发现 Visual Basic 可以从 Windows 95的多任务特征中获取许多好处。Visual Basic 4.0支持 OLE 2.0技术，允许用户从建立的应用程序中编辑数据，在有些情况下，用户甚至可以从 Visual Basic 应用程序编辑数据。在本章中，用户将学会怎样利用剪贴板交换信息，怎样用数据控件打开已存在的数据库，怎样寻找特定的项以及增加和删除数据库的记录。最后用户将学会用 OLE 控件和来自几个 Microsoft Office 应用程序的数据建立一个出版商信息系统。

12.1　剪贴板的应用

Windows 剪贴板允许用户在不同的 Windows 应用程序之间交换信息，它常常用于某个特定的 Windows 应用程序的剪切及粘贴操作。在同一时刻，剪贴板只能容纳一块同一类型的数据。如果将该格式的其他数据送入剪贴板，原来存储的数据就将被擦除掉。用户也可以通过下述方法清除剪贴板的内容：

```
Clipboard.Clear
```

用户还可以用 SetText 方法向剪贴板中送入数据，例如：

```
Clipboard.SetText data
```

该语句将数据 data（字符串数据）送入剪贴板中。

有时用户也需要从剪贴中取出当前文字的一个拷贝，这时用户可以 GetText 方法。GetText 方法将文本字符串从剪贴板上拷贝到字符串变量中，例如：

```
Temp $ =Clipboard.GetText(CF _ TEXT)
```

CF _ TEXT 为 Visual Basic 可识别的 Clipboard 的一种格式。虽然用 GetText 方法从剪贴板上取出了字符串，但剪贴板的内容不会有何变化，除非用户有意清除剪贴板或往剪贴板上送入新的信息。

在许多基于 Windows 的应用程序上都有块剪切和块粘贴等操作功能，例如字处理程序 Word，Visual Basic 中也有块剪切和块粘贴功能。为了能够编写出具有块剪切和块粘贴的 Windows 应用程序，用户应该对以下三条属性有所了解和掌握。

SelStart 属性

该属性适用于组合框和文本框，确定被选中文字的起始位置。如果其值为零，那么用户就是从文字的开始处选择。例如：

```
Text1.SelStart=Len(Text1.Text)/2
```

该语句将从文字中部开始选择文字。

SelLength 属性

该属性适用于组合框和文本框，它确定用户选中的字符数。如果 SelLength 为0，则没有选中任何文字。如果 SelLength 与字符串长度相等，则选中组合框或文本框的全部字符串。

SelText 属性

该属性也适用于组合框和文本框，它确定当前选定的字符串，如果用户没有选定字符，则它包含一个空串。当 SelText 的设置值为新值时，Visual Basic 用新赋的字符串代替被选定的文本，并且设置 SelLength 为零。

如果对于一个名为 Copy 的菜单项和一个名字为 Text1的文本框，只需编写以下代码就可以实现将文本框的内容拷贝到粘贴板上。

```
Sub Copy _ Click()
        Clipboard. SetText Text1. SelText
End Sub
```

如果用户想实现剪切选中的文字功能，只需要在上述代码中加入：

```
Text1. SelText=""
```

这样在复制被选中的文字之后把 SelText 属性的值设置为空字符串，就把被选中的文字剪切进剪贴板上了。

当用户把插入点移到文本框内，用下例代码就可实现由剪贴板向文本框粘贴信息的功能：

```
Sub Paste _ Click()
    Text1. Text=Clipboard. GetText()
End Sub
```

12.2　数据访问

12.2.1　数据控件

数据控件提供访问存储数据库中的数据的方法，并可以从一个记录移动到另一个记录，并同时在像文本框等约束控件中显示。如果没有数据控件，则在窗体中就不能有约束控件。

利用数据控件，用户可以不需要编写任何代码就执行大多数的数据访问操作。被约束于数据控件的控件自动显示当前记录的特定字段。如果数据控件移至别的记录，所有被约束的控件自动改为显示当前记录。如果用户改变数据，或从约束控件向数据控件输入新值，这些变化会自动存入数据库，此时数据控件移至新记录。

数据控件有两个用来访问数据库的属性：Database 和 RecordSet。

Database 属性

该属性是数据控件基本数据库对象的索引，在设计时不可用，运行时只读。

RecordSet 属性

该属性是对数据控件基本数据库的引用。

数据控件还常用到 DatabaseName 属性，它用来确定数据控件源数据的名字和位置。

12.2.2　显示数据库信息

用户通过 Visual Basic 工具箱的控件创建的对象都有显示数据库信息的嵌入能力。例如文本框、标签、组合框等。在数据库术语中，这些对象被称为约束控件。当一个对象的 DataSource 属性被设置为一个有效的数据库的名字，而且它的 DataField 属性被设置为在数据库中有效的段时，该对象被数据库所约束。一个 table 是一组域和记录，它是由用户或者他人在

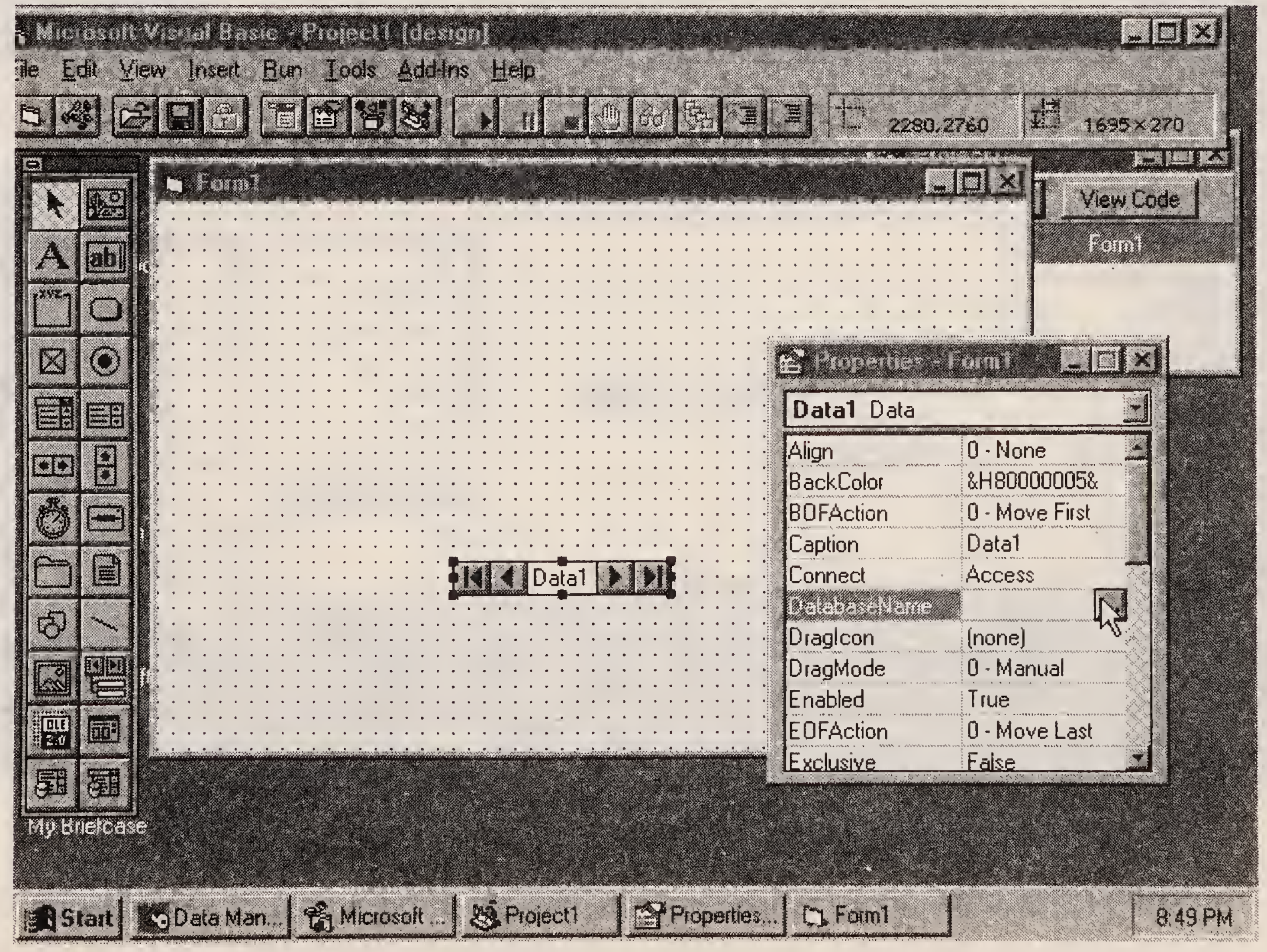
创建数据库时定义的。通过一个数据控件对象，用户能够把 Visual Basic 程序同数据库连接起来，这种联系建立之后，用户可以用由复选框、组合框和文本框等标准控件建立起来的对象显示数据库信息。

下面一个例子将用一个数据控件和四个文本框显示 BIBLIO. MDB 数据库的四个数据库域。这个例子说明了用户怎样创建一个数据库应用程序观看用户想查看的信息。在这个例子中，数据控件的 ReadOnly 属性被设置为 True，因此数据库中的信息只能被查看而不能被改变。如果用户想改变 BIBLIO. MDB 数据库的内容，则可以用属性窗口把 ReadOnly 属性设置为 False。为了防止原始的 BIBLIO. MDB 被改变，用户在操作下面步骤之前应该用资源管理器为它做一个备份。

（1）单击 File 菜单的 New Project 命令，建立新的工程文件。

（2）单击工具箱中的数据控件，把它加载到窗体上。打开属性窗口，选取 DatabaseName 属性并单击它右边的按钮，见图12.1。这时出现一对话框，用户可在此对话框中选取 BIBLIO. MDB 数据库，见图12.2。选取数据控件的 RecordSource 属性，单击它右边的下箭头打开列表，把 RecordSource 属性设置为 Title，见图12.3。用户还要设置数据控件的其他一些属性，它们的设置值见下表：

属性	设置
Caption	"BIBLIO. MDB"
Connect	Access
DatabaseName	"C:\vb\biblio. mdb"
Name	dBiblio
ReadOnly	True
RecordSetType	O-Table
RecordSource	Titles

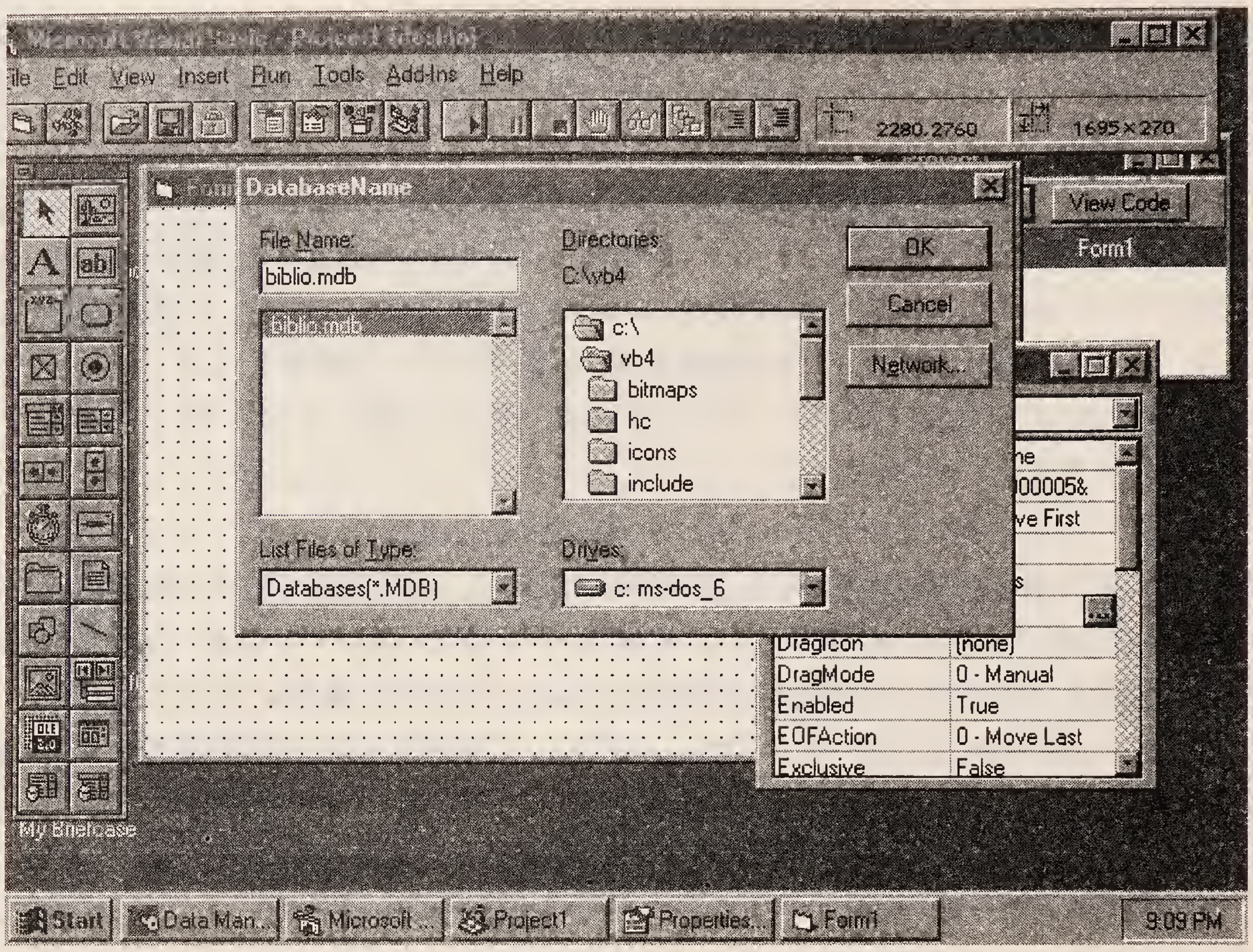

Microsoft Visual Basic - Project1 [design]
File
Edit
View
Insert
Run
Tools
Add-Ins
Help
2280,2760
1695×270
Form1
View Code
Form1
Data1
My Briefcase
Start
Data Man...
Microsoft ...
Project1
Properties...
Form1
8:49 PM

Properties - Form1

Data1 Data

Align	0 - None
BackColor	&H80000005&
BOFAction	0 - Move First
Caption	Data1
Connect	Access
DatabaseName	
DragIcon	(none)
DragMode	0 - Manual
Enabled	True
EOFAction	0 - Move Last
Exclusive	False

图12.1　打开属性设置对话框


DatabaseName
OK
Cancel
Network...
Start
Data Man...
Microsoft ...
Project1
Properties...
Form1
9:09 PM

File Name: biblio.mdb

Directories: C:\vb4

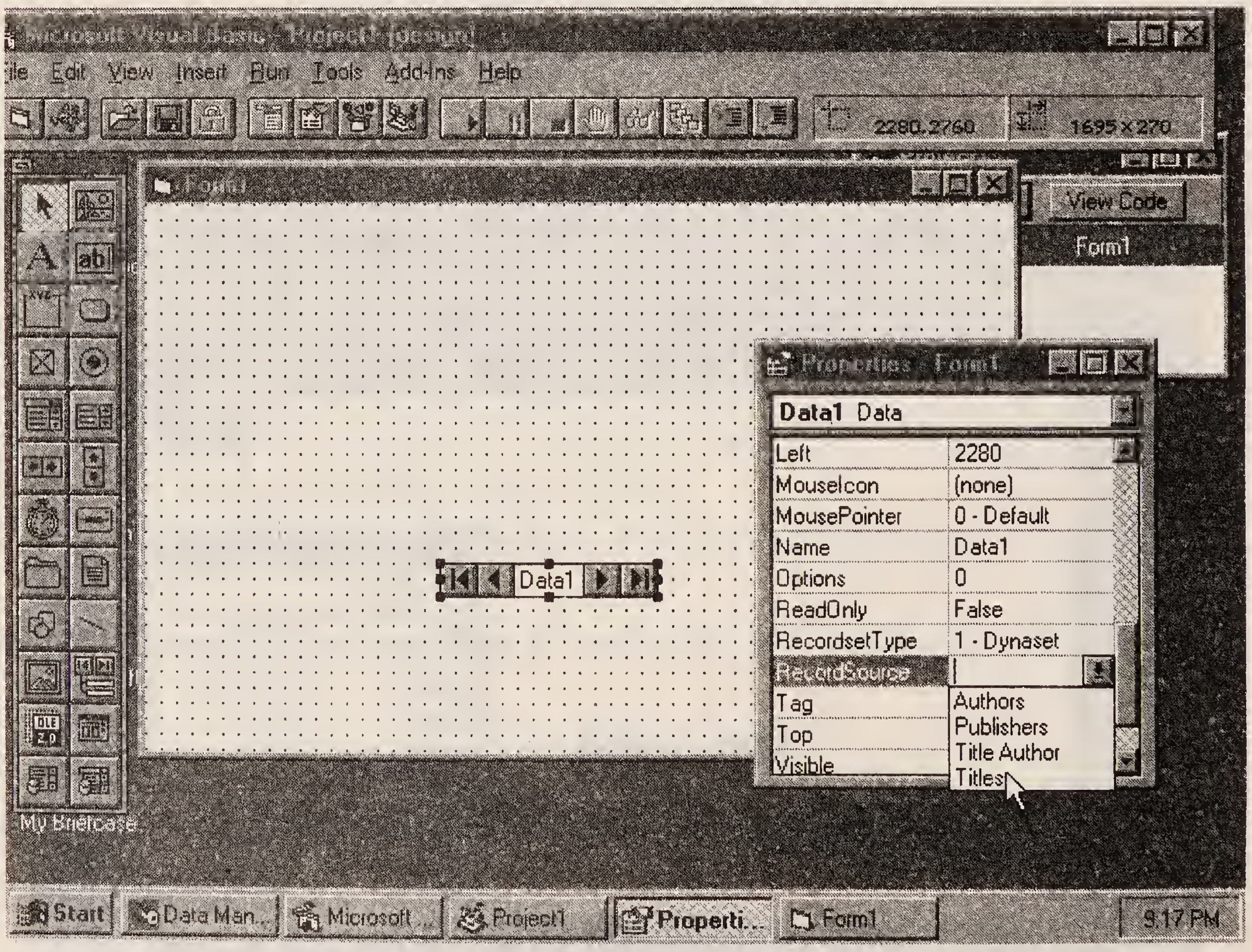
c:\
vb4
bitmaps
hc
icons
include

List Files of Type: Databases(*.MDB)

Drives: c: ms-dos_6

DragIcon	(none)
DragMode	0 - Manual
Enabled	True
EOFAction	0 - Move Last
Exclusive	False

图12. 2　设置属性


Form1
View Code
Properties - Form1
Data1
Start
Data Man...
Microsoft ...
Project1
Properti...
Form1
9:17 PM

Data1 Data

Left	2280
MouseIcon	(none)
MousePointer	0 - Default
Name	Data1
Options	0
ReadOnly	False
RecordsetType	1 - Dynaset
RecordSource	Authors Publishers Title Author Titles
Tag	
Top	
Visible	

图12. 3　设置 RecordSource 属性

(3) 将文本框、标签和命令按钮等控件增加到窗体，该文本框将用来显示来自数据库的信息，用户必须把它约束给数据控件。为了做到这一点，用户只需把文本框的 DataSource 属性设置为数据控件的名字，此处为 dBiblio。接下来将设置文本框将显示的字段，单击 DataField 属性右边的下箭头，这时显示出文本框能够获取的当前数据源的字段，见图12.4。文本框和标签的属性设置见下表。

对象	属性	设置
文本框1	DataField	Title
	DataSource	dBiblio
	Name	TxtTitle
	Text	空白
标签1	Caption	BookTitle
	Name	Label1
文本框2	DataField	ISBN
	DataSource	dBiblio
	Name	TxtISBN
	Text	空白
标签2	Caption	ISBN
	Name	Label2
文本框3	DataField	YearPublished
	DataSource	dBiblio
	Name	txtYear
	Text	空白
标签3	Caption	Year
	Name	Label3
命令按钮	Name	CmQuit
	Caption	Quit

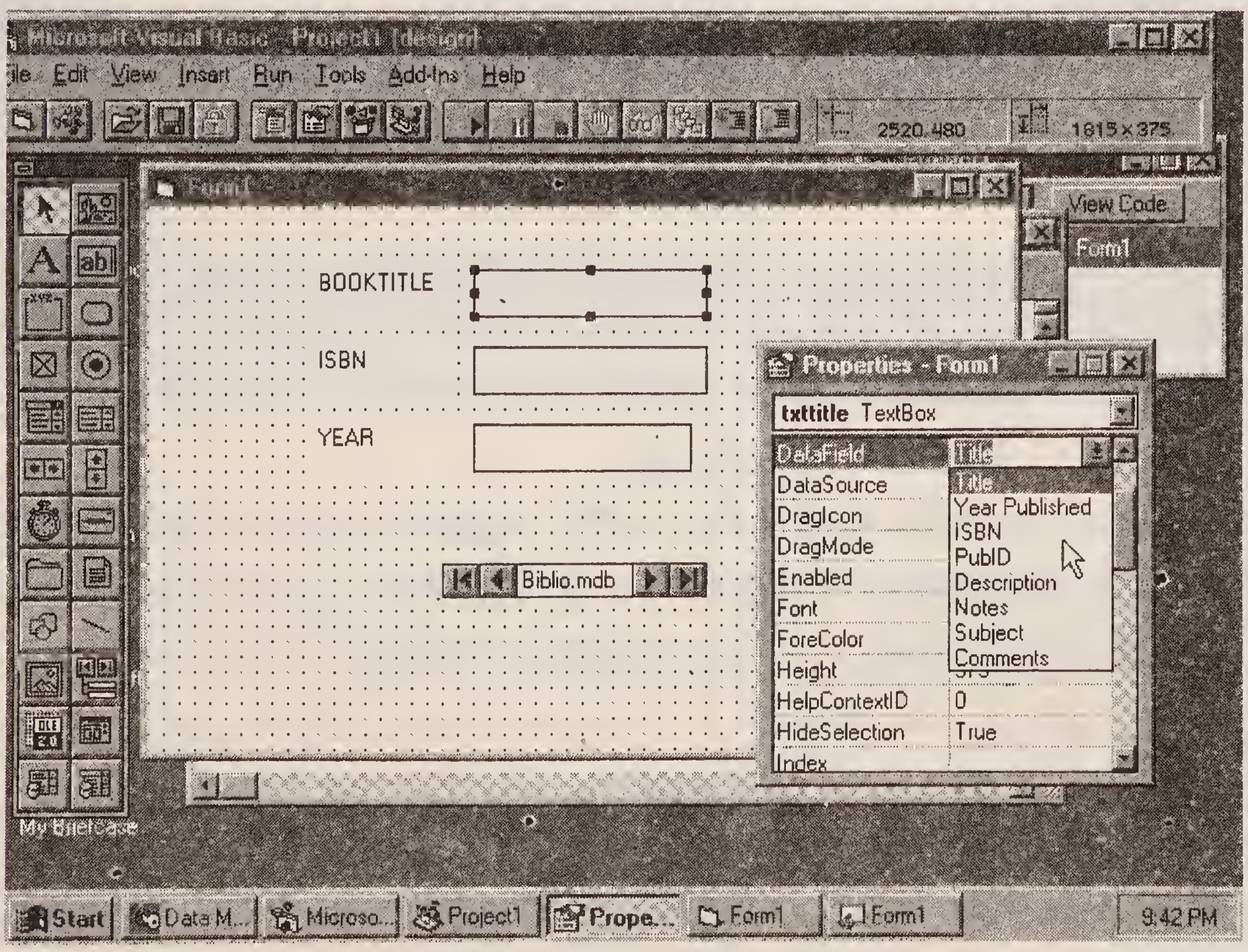

Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
2520,480
1815 x 375
Form1
BOOKTITLE
ISBN
YEAR
Biblio.mdb
View Code
Form1
Properties - Form1
txttitle TextBox
DataField
Title
DataSource
DragIcon
DragMode
Enabled
Font
ForeColor
Height
HelpContextID
0
HideSelection
True
Index
Title
Year Published
ISBN
PubID
Description
Notes
Subject
Comments
My Briefcase
Start
Data M...
Microso...
Project1
Prope...
Form1
Form1
9:42 PM

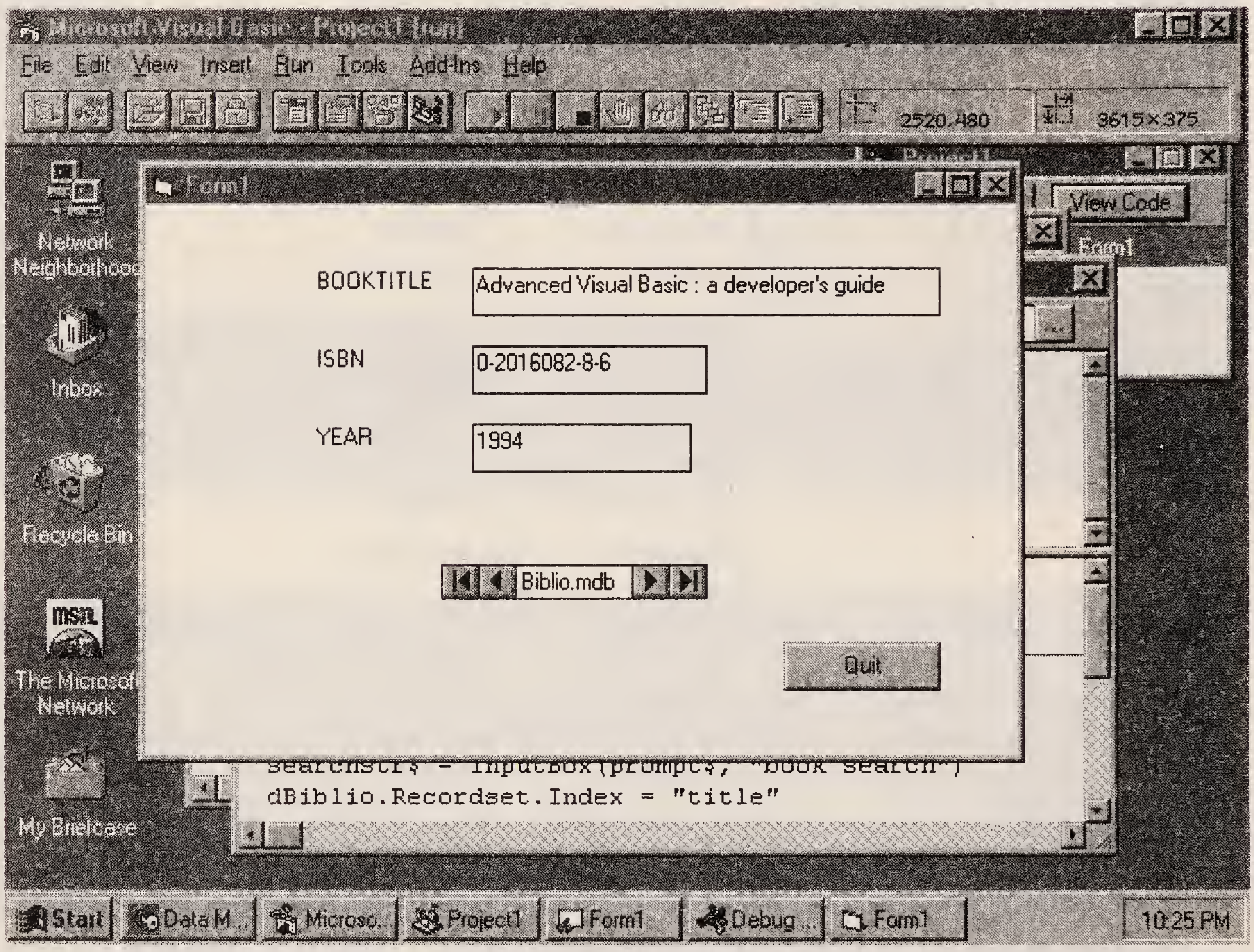
图12.4　打开数据源的字段

从上表用户可以看到三个文本框的 DataSource 属性都被设置为 dBiblio，这样它们都被约束给数据控件。

(4) 双击 Quit 按钮打开代码窗口，键入以下代码：

```
Sub CmQuit()
   End
End Sub
```

(5) 单击工具条上的 Start 按钮或按 F5键运行程序，BIBLIO. MDB 数据库中的 Title 表的第一个记录出现在窗体中，见图12.5。


Microsoft Visual Basic - Project1 [run]
File Edit View Insert Run Tools Add-Ins Help
2520,480
3615×375
View Code
Form1
Form1
Network Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Biblio.mdb
Quit
Start
Data M...
Microso...
Project1
Form1
Debug...
Form1
10:25 PM

BOOKTITLE　Advanced Visual Basic : a developer's guide

ISBN　0-2016082-8-6

YEAR　1994

```
searchstr$ = inputbox(prompt$, "book search")
dBiblio.Recordset.Index = "title"
```

图12.5　显示记录

(6) 单击数据控件右边的第一个箭头查看 Title 表的第二个记录，用户可以注意到三个文本框内容都被更新。

(7) 单击数据控件右边的第二个箭头查看 Title 表的最后一个记录。

(8) 单击 Quit 命令按钮结束程序的运行。

这个程序仅含有一行代码，但是它给了用户一定量的信息。关于这个程序最值得一讲的是它仅显示了 Biblio 数据库中用户想查看的字段。现在用户已经看到了，用一个数据控件和几个数据约束文本框就能够创建一个有效的数据库应用程序。

12.2.3　寻找记录

Visual Basic 提供了一个 RecordSet 对象，该对象代表着在数据库中用户正在程序中使用的那部分数据。RecordSet 对象包含着特殊的属性和方法，用户可以用这些属性和方法寻找、排序、增加和删除记录，在下面的例子中，用户将用 RecordSet 对象在 BIBLIO. MDB 数据库中寻找书的标题。

在工具框中单击命令按钮，把它增加到窗体的左下端，设置它的 Caption 和 Name 属性分别为 Search 和 CmSearch，见图12.6。双击 Search 按钮，在代码窗口打开 CmSearch _ Click ()事件过程，键入如下代码：

```
Sub CmSearch _ Click()
   Prompt $ ="Enter the full booktitle"
   SearchStr $ =InputBox(prompt $ ,"Book Search")
   dBiblio. RecordSet. Index="Title"              'useTitleTable
   dBiblio. RecordSet. Seek "=",SearchsTr $       'and Search
   If dBiblio. RecordSet. NoMatch Then
        MsgBox("Sorry,I couldn't find your book")
        dBiblio. RecordSet. MoveFirst             'go to first record
   End If
End Sub
```

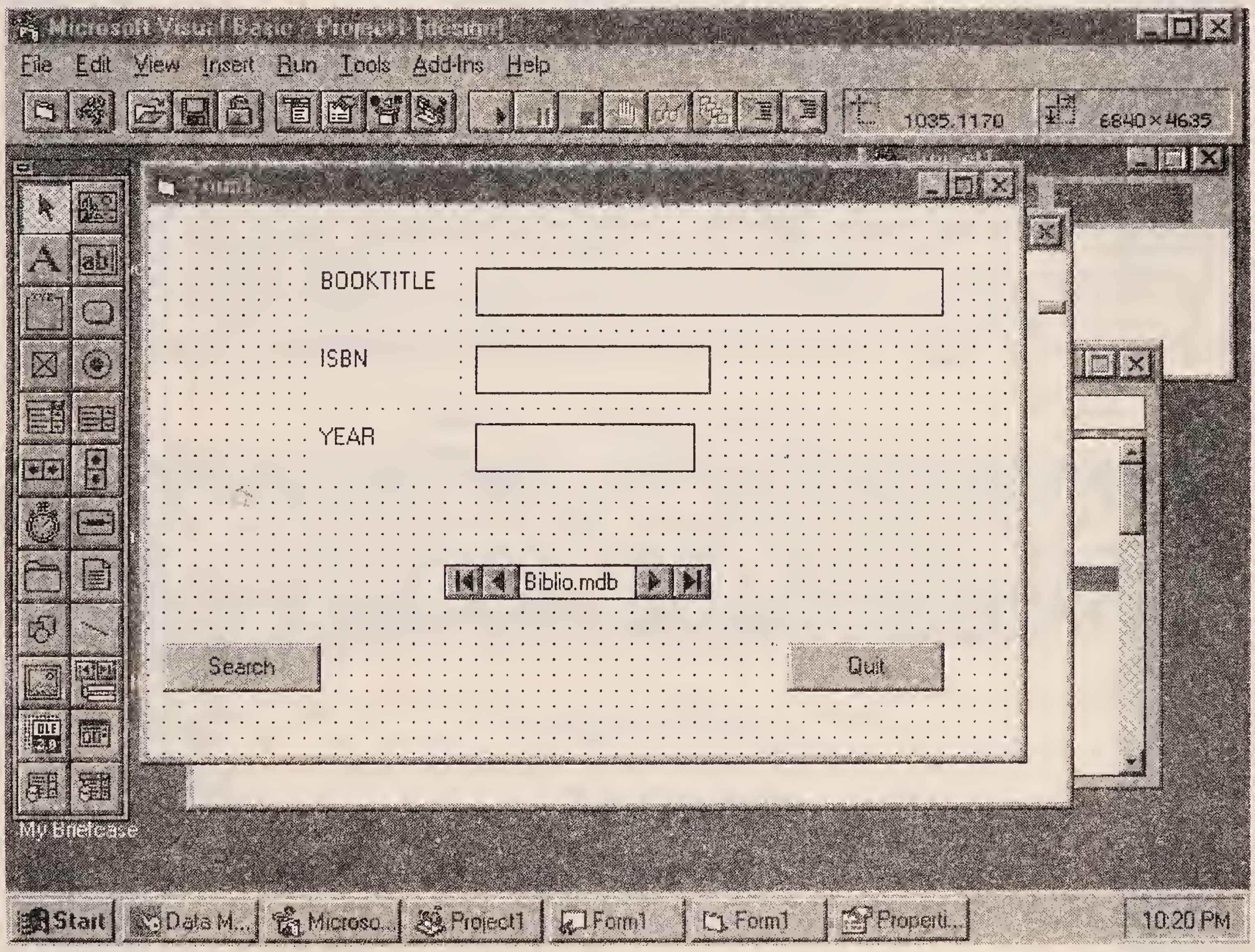
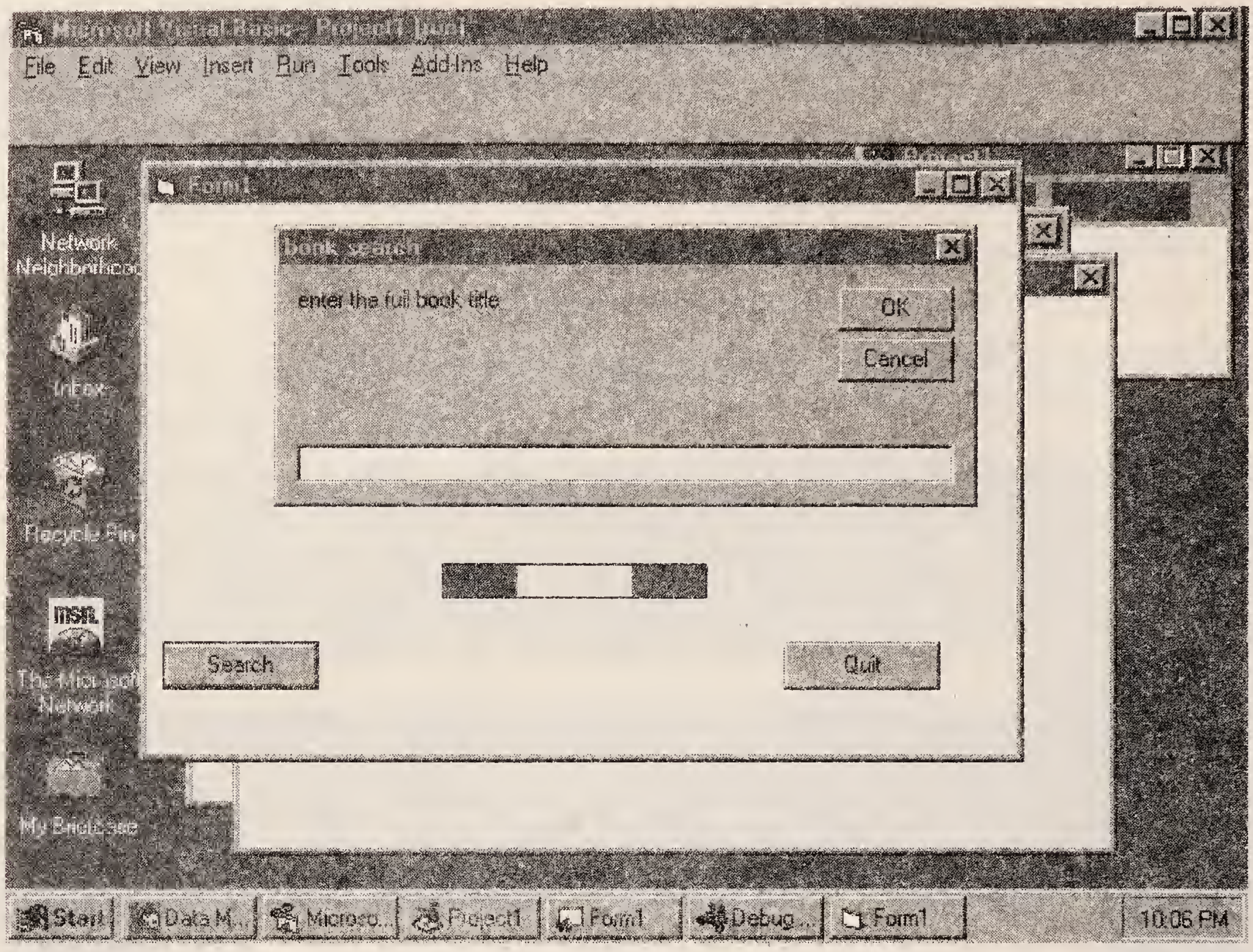

Microsoft Visual Basic - Project1 [design]
File
Edit
View
Insert
Run
Tools
Add-Ins
Help
1035,1170
6840×4635
Form1
BOOKTITLE
ISBN
YEAR
Biblio.mdb
Search
Quit
My Briefcase
Start
Data M...
Microso...
Project1
Form1
Form1
Properti...
10:20 PM

图12.6　增加寻找按钮

在上述代码中，Index 属性用来确定将被寻找的数据库字段，Seek 方法用来寻找记录，后面附加“=”，用来比较寻找的字符串和数据库中的文本，与此相关的操作还有>=，>，<=和<。Seek 方法寻找数据库的 Title 字段直到发现一个相匹配的字符串或到达列表的结尾。如果没有发现匹配的字符串，屏幕上将显示一个消息框，并且出现第一个记录。如果在寻找过程中没有匹配的字符串被发现，NoMatch 属性被设置为 True。MoveFirst 方法使第一个记录成为当前记录。

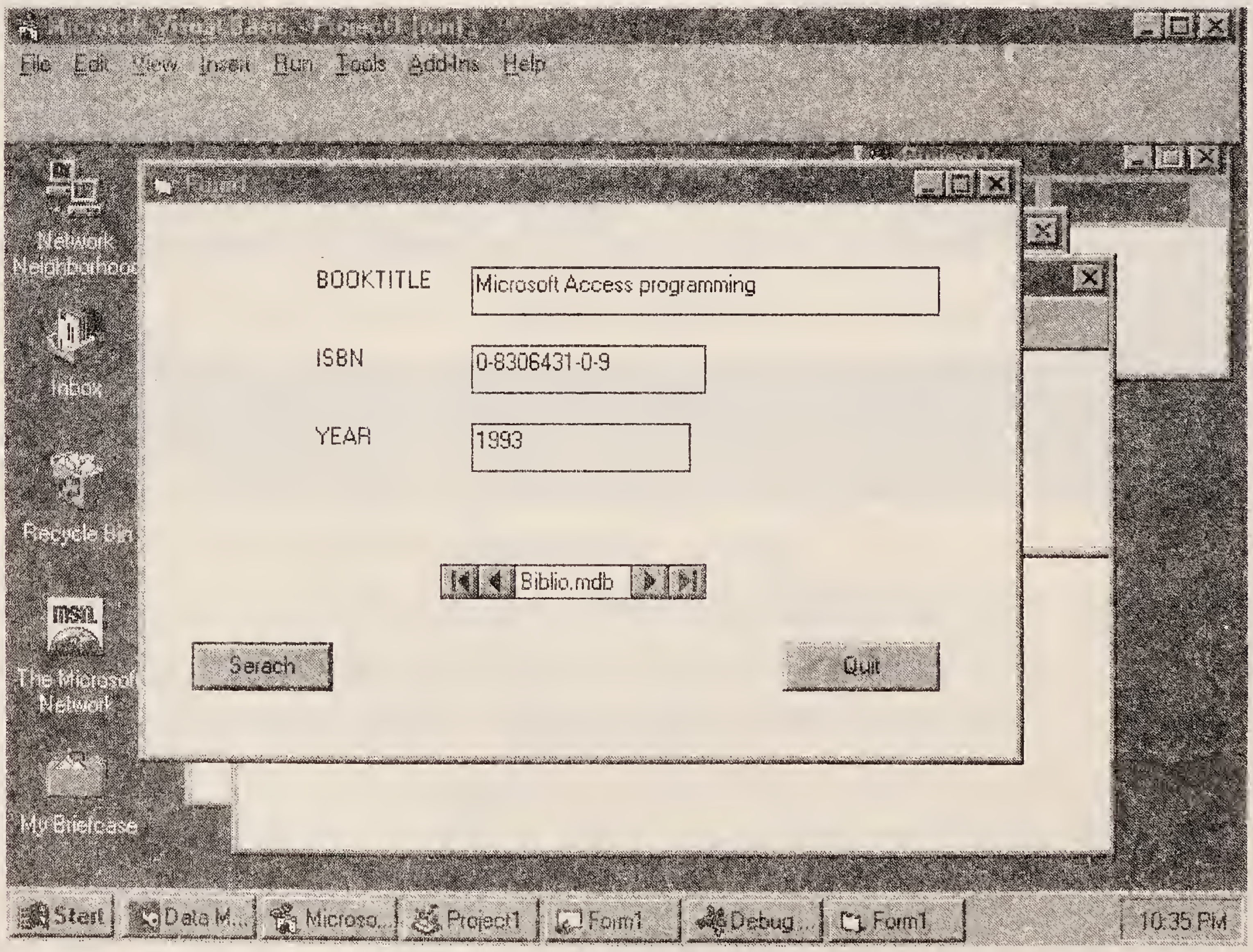
用户在保存程序以后，可以通过按 F5键或单击工具条的 Start 图标来运行程序。像以前一样，BIBLIO. MDB 数据库的 Title 表的信息出现在文本框中。单击 Search 按钮，这时出现一个寻找对话框，见图12.7。


Microsoft Visual Basic - Project1 [run]
File Edit View Insert Run Tools Add-Ins Help
Form1
book search
OK
Cancel
Search
Quit
Network Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Start
Data M...
Microso...
Project1
Form1
Debug...
Form1
10:06 PM

enter the full book title

图12.7　寻找对话框

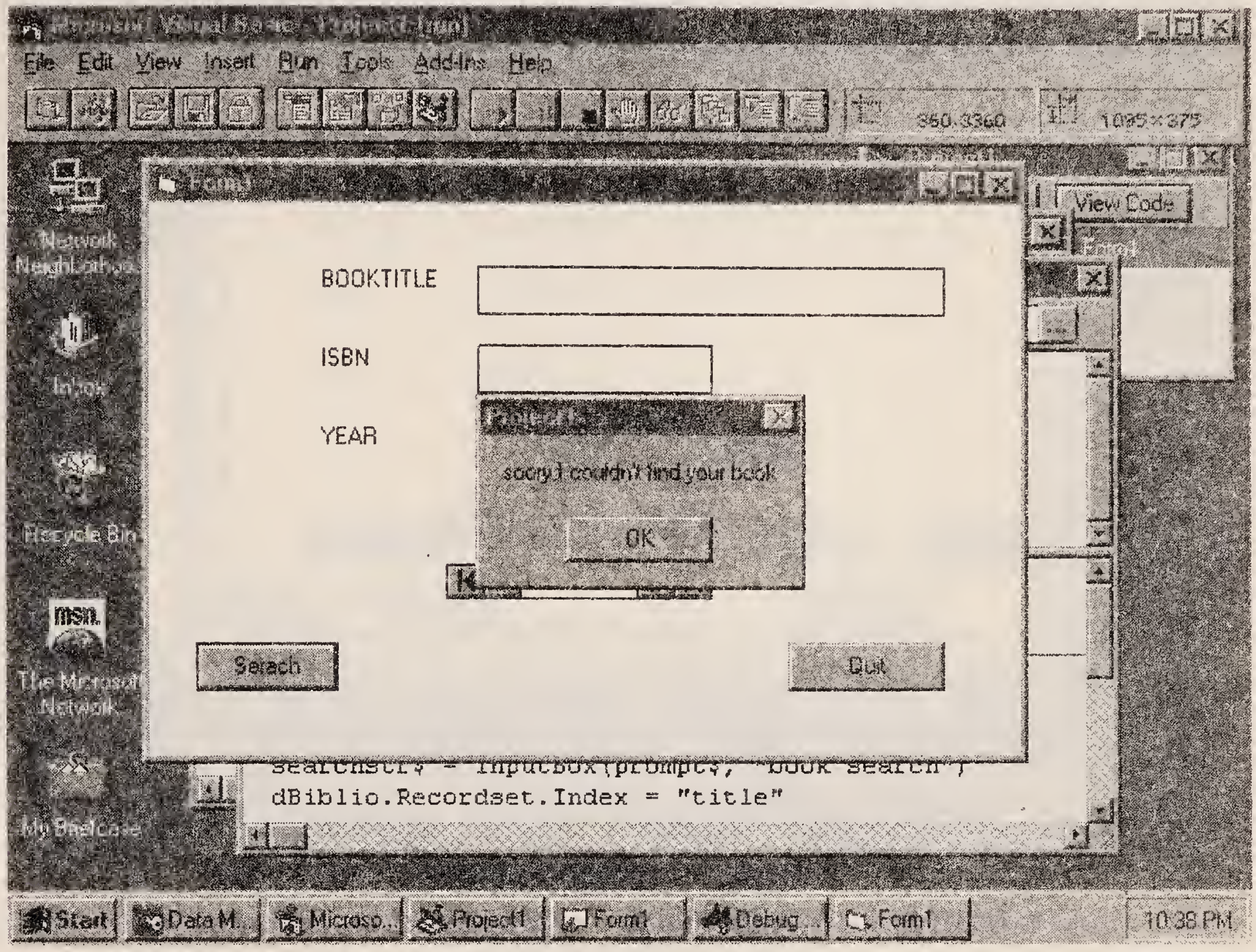
用户可以在寻找对话框中键入要寻找的信息，例如键入 Microsoft Access Programming，单击 OK，如果它存在，则寻找到时显示图12.8的画面。


Microsoft Visual Basic - Project1 [run]
File Edit View Insert Run Tools Add-Ins Help
Form1
Biblio.mdb
Serach
Quit
Network Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Start
Data M...
Microso...
Project1
Form1
Debug...
Form1
10:35 PM

BOOKTITLE　Microsoft Access programming
ISBN　0-8306431-0-9
YEAR　1993

图12.8　显示寻找到的信息

如果没有寻找到有关信息时会出现一个消息框，例如键入 WWW 后按回车，则出现图12.9所示的消息框，单击 OK 按钮可关闭消息框，单击 Quit 退出程序。


File
Edit
View
Insert
Run
Tools
Add-Ins
Help
360,3360
1095×375
View Code
BOOKTITLE
ISBN
YEAR
OK
Serach
Quit
Network Neighborhood
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Start
Data M...
Microso...
Project1
Form1
Debug...
Form1
10:38 PM

sorry,I couldn't find your book

```
dBiblio.Recordset.Index = "title"
```

图12.9　显示没有找到消息

12.2.4　增添记录

为了增添一个新记录到数据库中，用户要把数据控件的 ReadOnly 属性设置为 False，然后用户可以在相关事件过程中用 AddNew 方法在数据库中打开一个新记录。当有一个空白记录出现在窗体上时，用户可以在需要的字段中填写，然后用户可以通过单击数据控件上的箭头，移到数据库的另一个记录，这样新的记录按字母顺序插入到数据库。

下面的例子将说明怎样用 AddNew 方法在数据库中插入一个新记录。

（1）单击工具箱中的命令按钮，把它增加到窗体的底部，设置它的 Name 和 Caption 属性分别为 CmAdd 和 Add，见图12.10。

（2）双击 Add 按钮，在 CmAdd _ Click()事件过程中键入以下代码：

```
Sub CmAdd _ Click()
   Str1 $ ="Enter new record"
   Str2 $ =MsgBox(Str1 $ ,vbokCancel,"Add Record")
   If Str2 $ =vbok Then
       txtTitle. SetFocus
       dBiblio. RecordSet. AddNew
       dBiblio. RecordSet. Feilds("PubID")="14"
   End If
End Sub
```

（3）单击工具条上的 Start 图标或按 F5键运行程序。

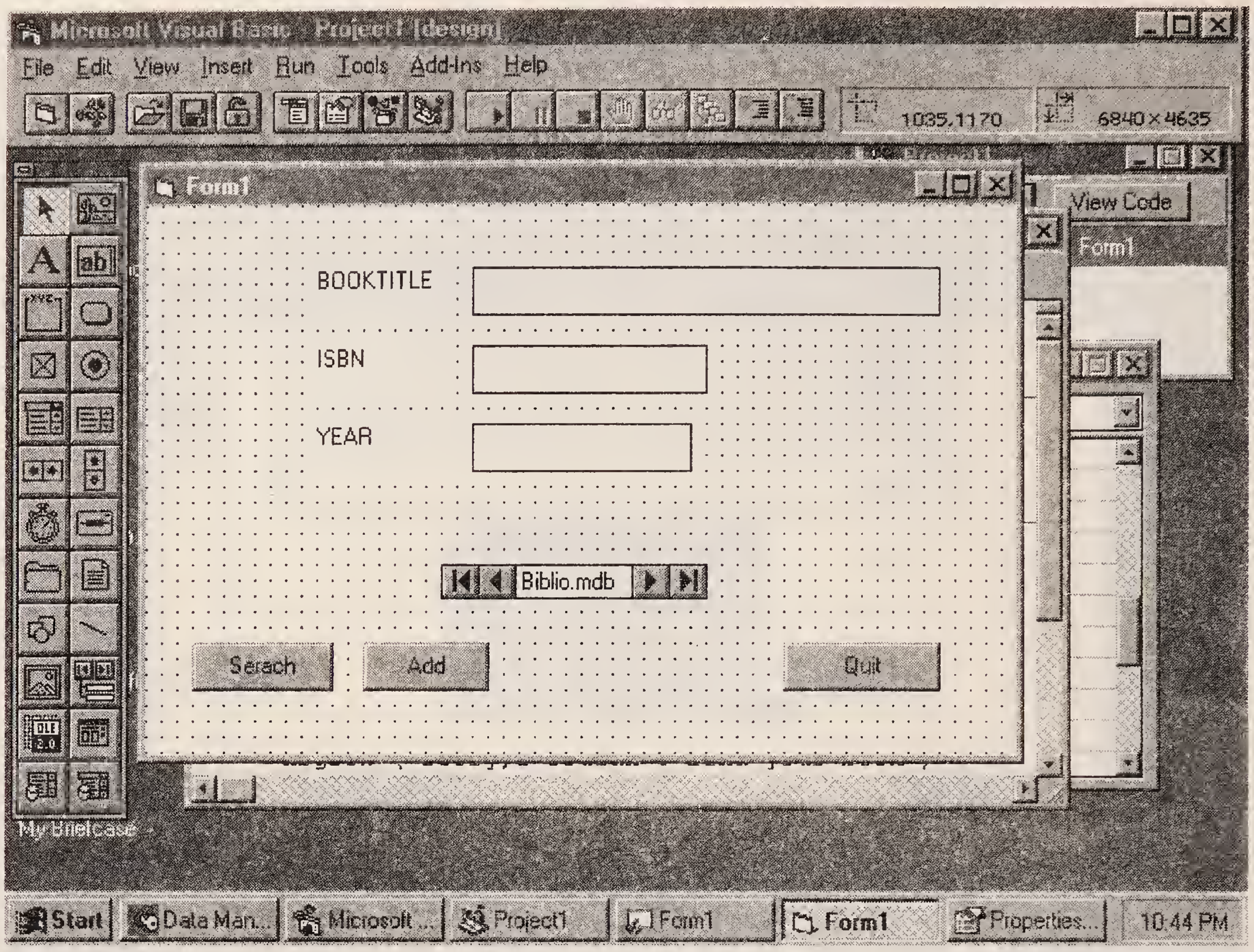
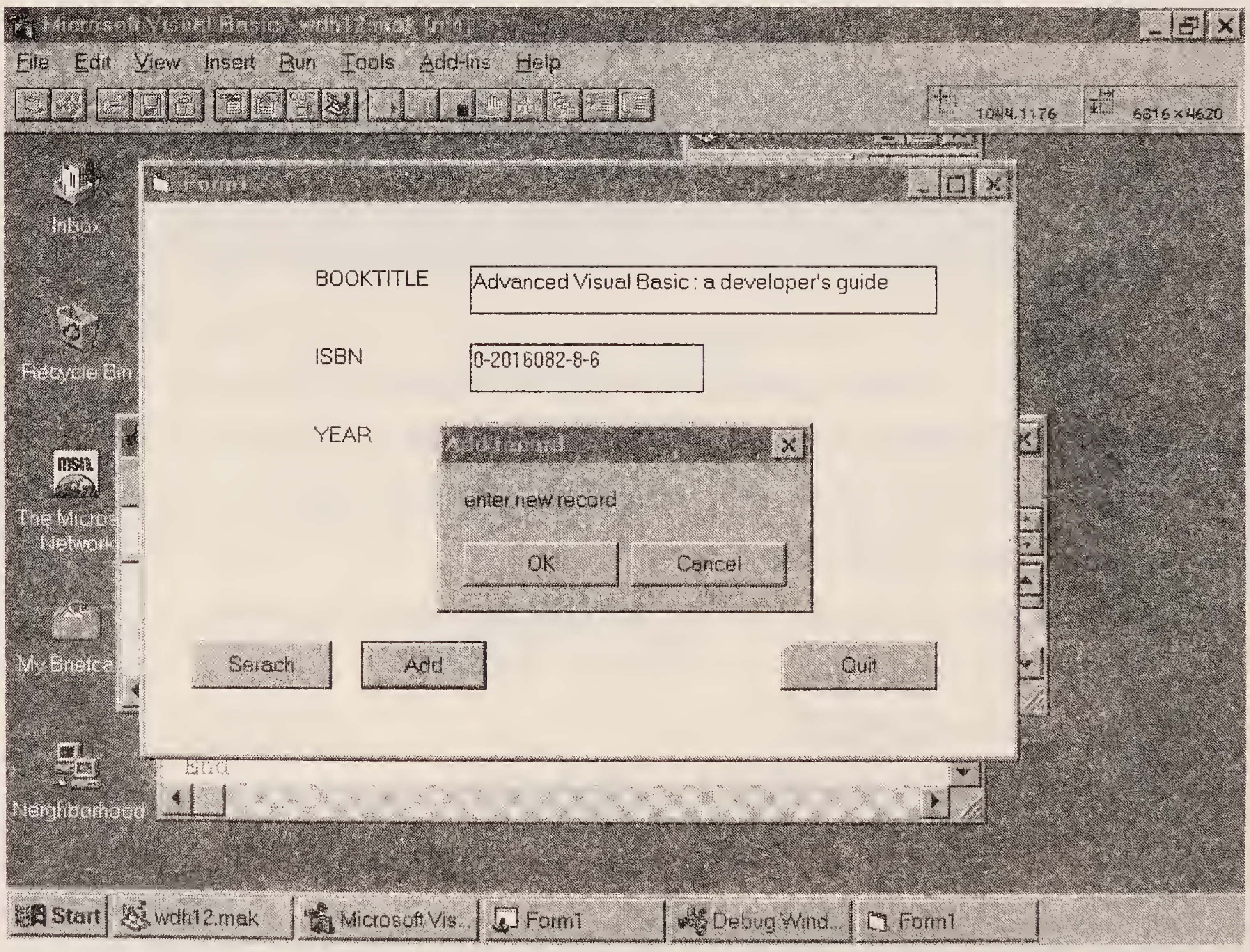

Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
1035,1170
6840 x 4635
Form1
BOOKTITLE
ISBN
YEAR
Biblio.mdb
Serach
Add
Quit
View Code
Form1
My Briefcase
Start
Data Man...
Microsoft ...
Project1
Form1
Form1
Properties...
10:44 PM

图12. 10　添加 Add 按钮

(4) 单击 Add 按钮，这时一个对话框出现，见图12. 11。


Microsoft Visual Basic - wdh12.mak [run]
File Edit View Insert Run Tools Add-Ins Help
1044,1176
6816 x 4620
Form1
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Network Neighborhood
Add record
OK
Cancel
Serach
Add
Quit
Start
wdh12.mak
Microsoft Vis...
Form1
Debug Wind...
Form1

BOOKTITLE　Advanced Visual Basic : a developer's guide

ISBN　0-2016082-8-6

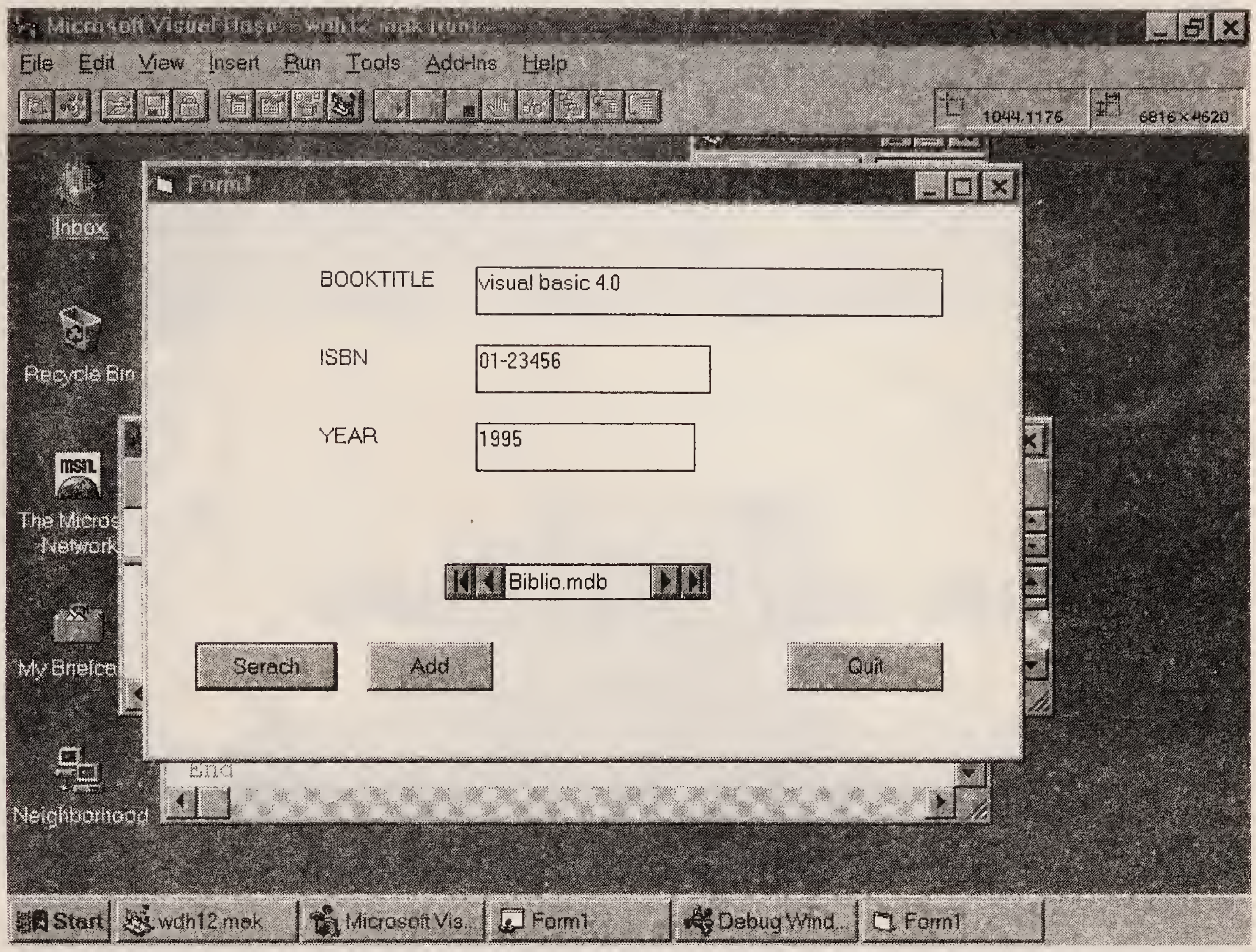
YEAR

enter new record

图12. 11　询问对话框

(5) 单击 OK 按钮，这时一个新的、空白的记录出现在窗体上，用户可在里面键入新的记录（可用 TAB 键在不同字段之间切换），例如键入：

Visual Basic 4.0，01—23456，1995，见图12.12。



BOOKTITLE visual basic 4.0

ISBN 01-23456

YEAR 1995

File Edit View Insert Run Tools Add-Ins Help
1044,1176
6816×4620
Form1
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Neighborhood
Biblio.mdb
Serach
Add
Quit
End
Start
wdh12.mak
Microsoft Vis...
Form1
Debug Wind...
Form1

图12.12　键入新记录

(6) 单击数据控件的左外箭头，这时新记录被插进数据库中。

(7) 单击 Quit 按钮退出程序。

用户可以用 Add 按钮加任意数量的记录到 BIBLIO.MDB 数据库中。

12.2.5　删除记录

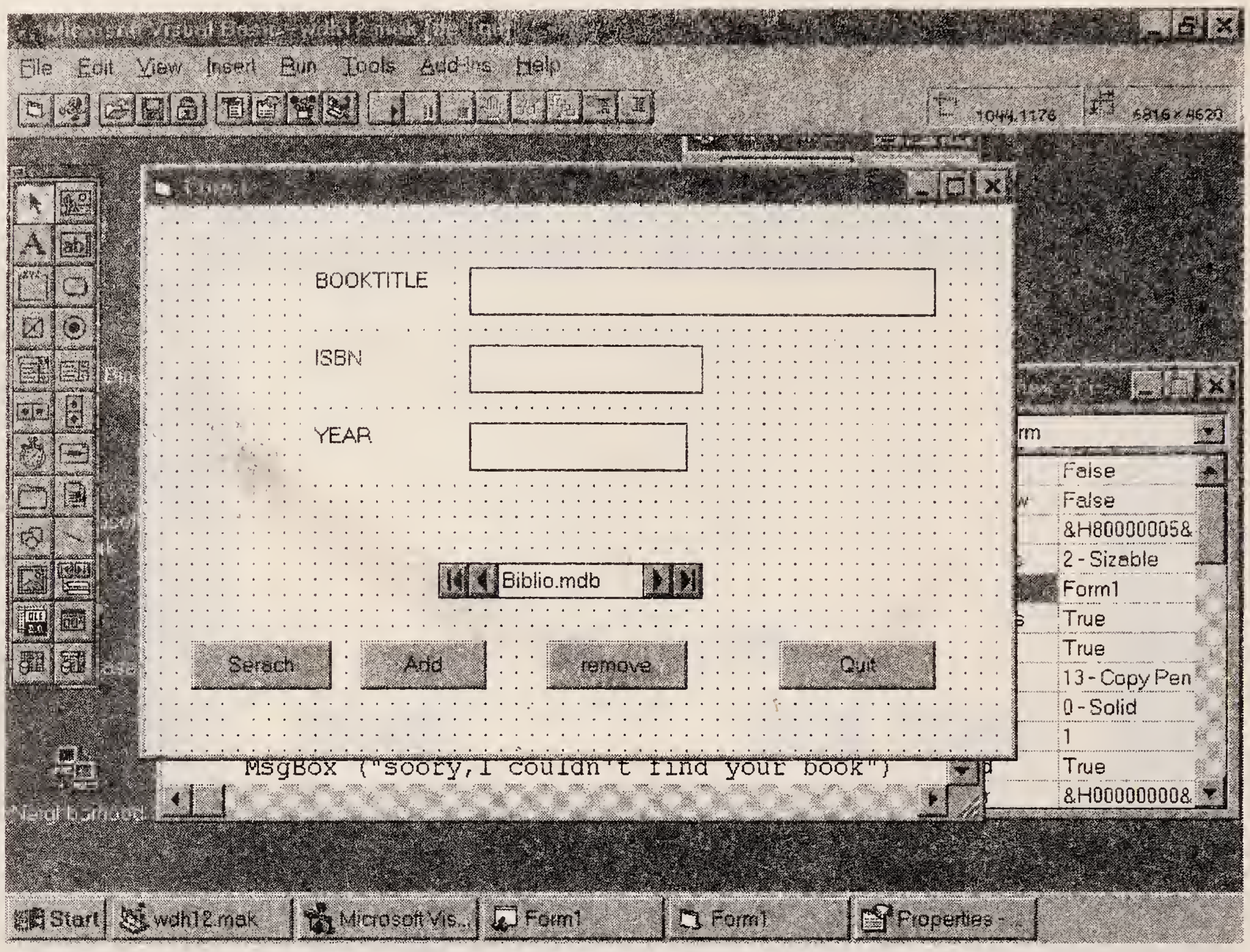
为了删除一个记录，用户应先显示想删除的记录，然后用 RecordSet 对象的删除方法删除一个记录。在删除一个记录时，ReadOnly 属性也应该被设置成 False。当用户删除了一个记录之后应该显示数据库中的另一个记录，最常用的方法是用 MoveFirst 方法显示数据库中的第一个记录。

用户可按下面的步骤删除一个记录：

(1) 单击工具箱的命令按钮，把它增加到窗体的底部。设置它的 Name 和 Caption 属性分别为 CmRemove 和 Remove，见图12.13。

(2) 双击 Remove 按钮打开代码窗口的 CmRemove _ Click()事件过程，键入以下代码：

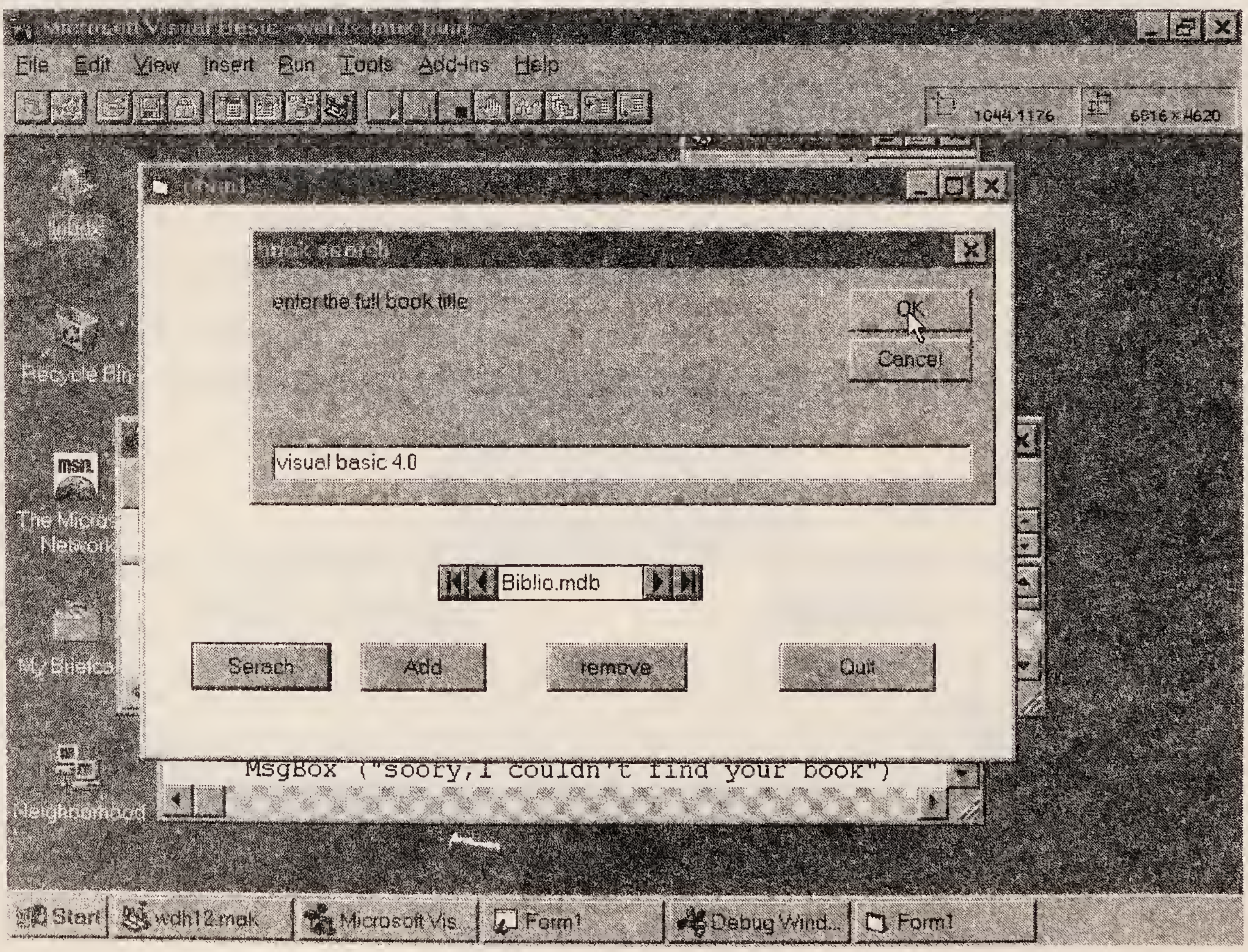
```
Sub CmRemove _ Click()
    Strl $ ="Do you really want to delete a record"
    Str2 $ =MsgBox(Str1 $ ,vbYesNo,"Remove record")
    If Str2 $ =vbYes Then
            dBiblio.RecordSet.Delete
            dBiblio.RecordSet.Movefirst
    End If
End Sub
```


File Edit View Insert Run Tools Add-Ins Help
1044,1176
6816 x 4620
BOOKTITLE
ISBN
YEAR
Biblio.mdb
Serach
Add
remove
Quit
False
False
&H80000005&
2 - Sizable
Form1
True
True
13 - Copy Pen
0 - Solid
1
True
&H00000000&
Start
wdh12.mak
Microsoft Vis...
Form1
Form1
Properties -

```
MsgBox ("soory,I couldn't find your book")
```

图12.13　添加删除按钮

(3) 单击工具条上的 Start 图标或按 F5键运行程序，单击 Search 按钮寻找用户上次增加的记录，见图12.14。单击 OK，上次添加的记录出现。


File Edit View Insert Run Tools Add-Ins Help
1044,1176
6816 x 4620
OK
Cancel
Biblio.mdb
Serach
Add
remove
Quit
Recycle Bin
Neighborhood
Start
wdh12.mak
Microsoft Vis...
Form1
Debug Wind...
Form1

enter the full book title

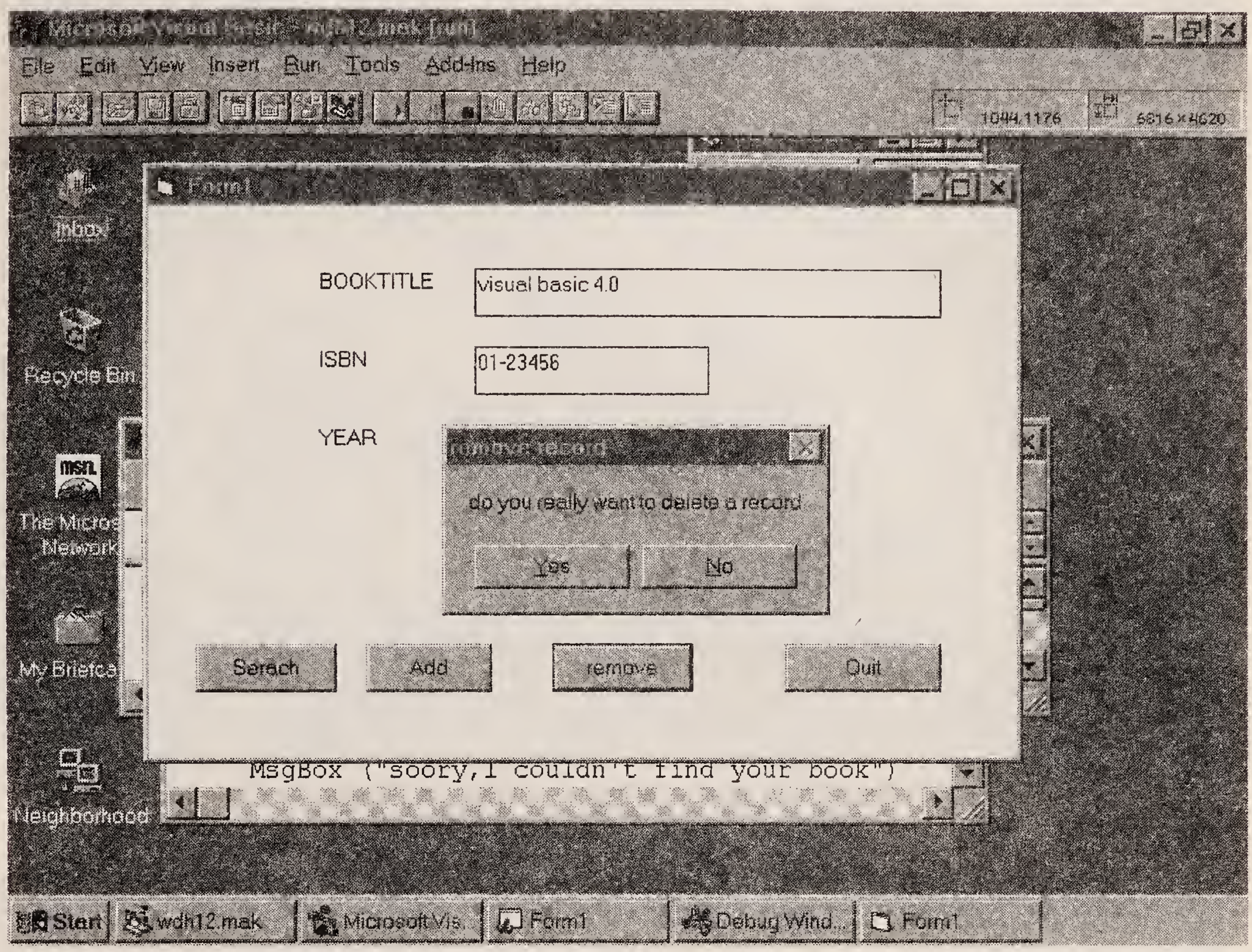
visual basic 4.0

```
MsgBox ("soory,I couldn't find your book")
```

图12.14　寻找记录

（4）单击 Remove 按钮，这时出现一个消息框询问用户是否真的想删除一个记录，见图12.15。如果想删除，单击 OK，这时被显示的记录被删除。

（5）单击 Quit 按钮退出程序。


File
Edit
View
Insert
Run
Tools
Add-Ins
Help
1044,1176
6816×4620
Form1
Inbox
Recycle Bin
The Microsoft Network
My Briefcase
Neighborhood
Yes
No
Serach
Add
remove
Quit
Start
wdh12.mak
Microsoft Vis...
Form1
Debug Wind...
Form1

BOOKTITLE visual basic 4.0

ISBN 01-23456

YEAR

do you really want to delete a record

```
MsgBox ("soory,I couldn't find your book")
```

图12.15　删除记录

12.3　数据管理器

数据管理器是创建由 Visual Basic 标准版提供数据库的唯一方式，它是一种独立的应用程序，它能够帮助用户创建一个新的 Visual Basic/Access 数据库。单击 Add _ Ins 菜单下的 Data Manager... 命令可以打开数据管理器，它的主屏幕见图12.16。

数据管理器允许用户以各种形式查看现有数据库的结构，用户能够用数据管理器创建和查看的各种形式的数据库列在下表：

数据库形式	支持
Microsoft Access	创建、查看
Paradox 3.x and 4.x	查看
dBASE III and IV	查看
FoxPro 2.0 and FoxPro 2.5	查看
Btrieve	查看

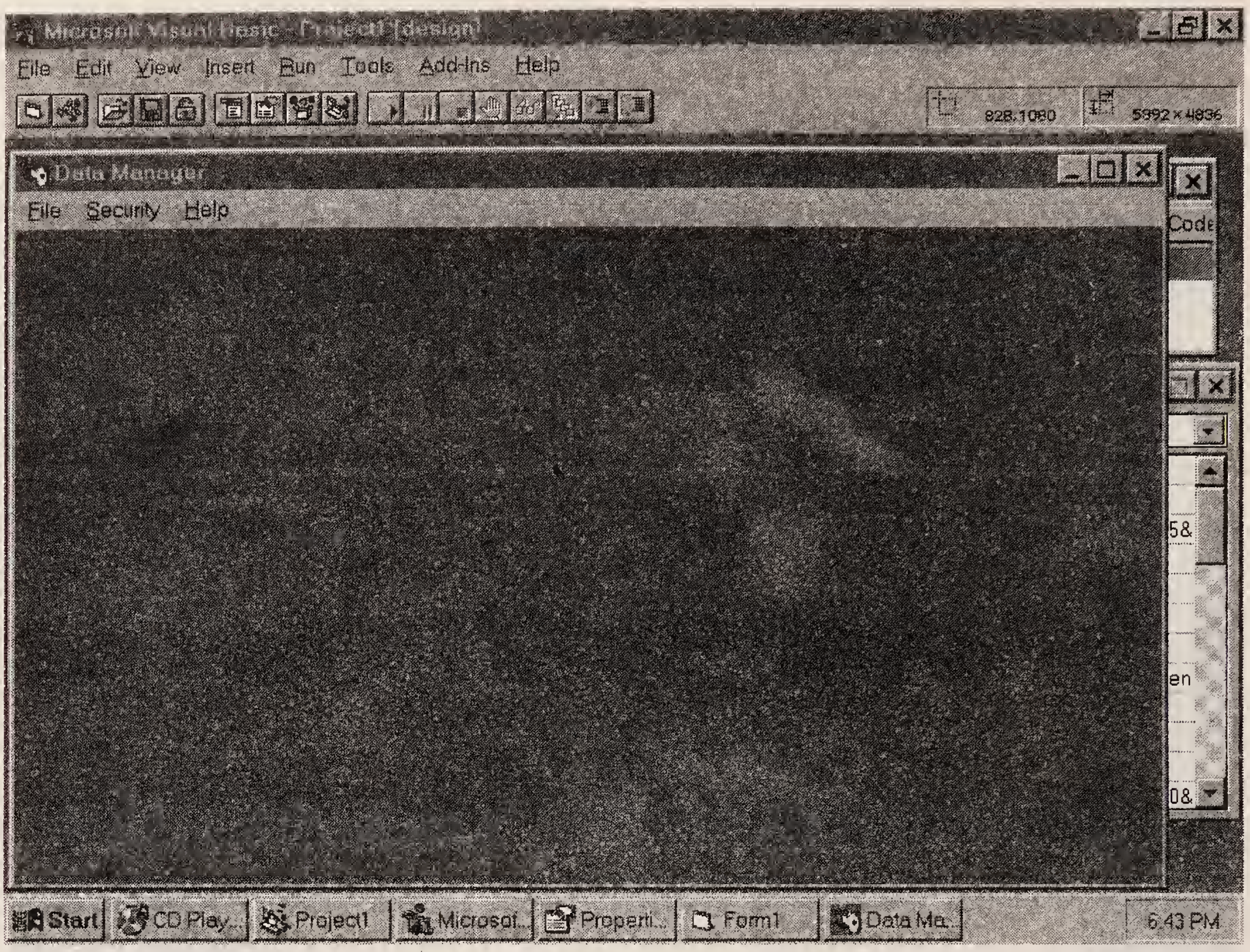

Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
828,1080
5992 x 4836
Data Manager
File Security Help
Code
Start
CD Play...
Project1
Microsof...
Properti...
Form1
Data Ma...
6:43 PM

图12.16　数据管理器的主屏幕

12.3.1　创建新的数据库

如果用户想创建一个新的 Visual Basic 数据库，则用户没有格式的选择，创建的数据库类型是能够被 Microsoft Access Version 2.0打开和操作的新数据库。

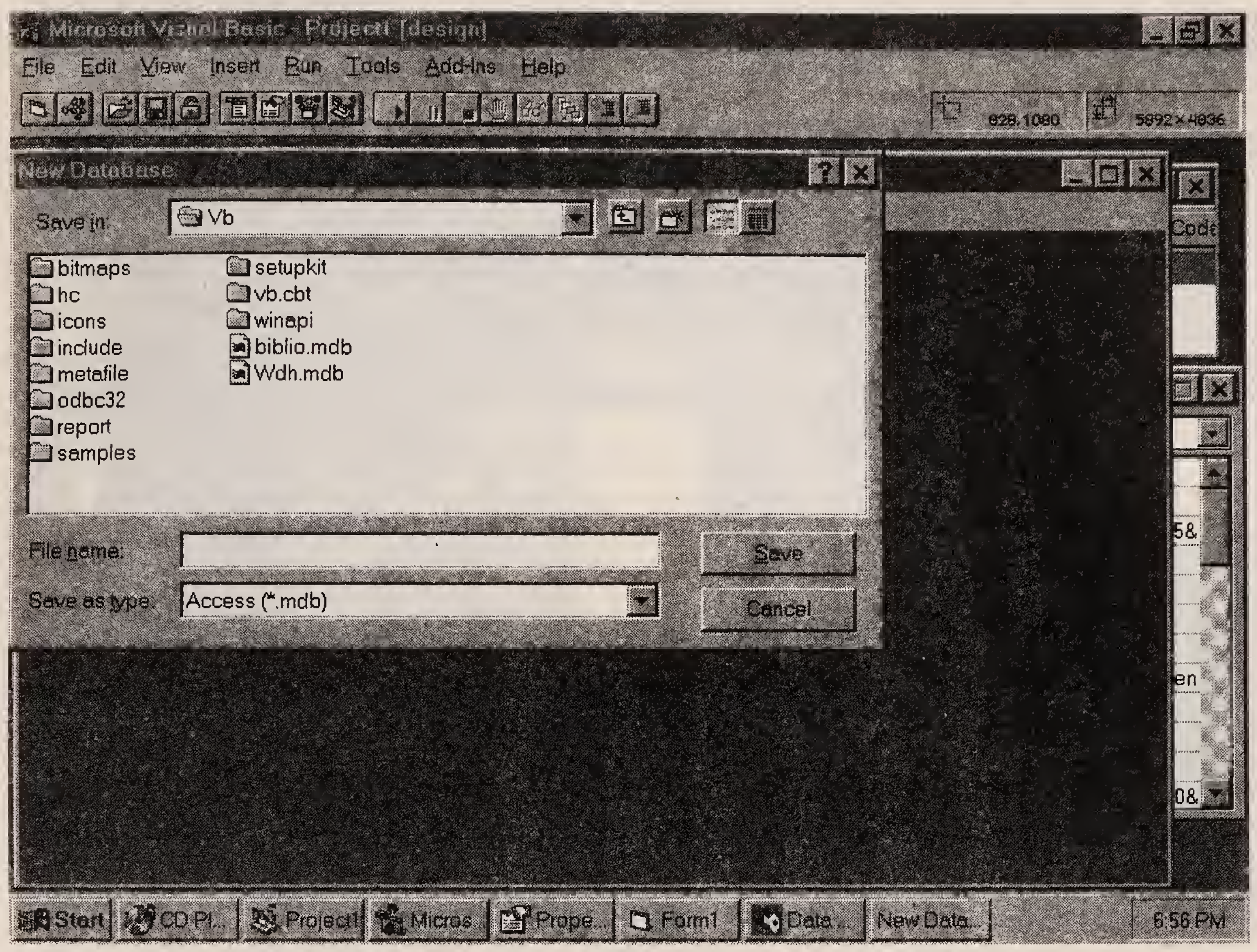
用户想创建新的数据库，可单击数据管理器 File 菜单下的 New Database 命令，这时出现一个建立新的数据库的对话框，见图12.17。

用户可以在对话框中键入存放新数据库的目录和文件名——文件名的扩展名必须是 MDB，用户不能键入或选择已存在数据库的文件名，单击 Save 按钮保存数据库的文件名。例如，用户可键入 C:\VB\MYBASE，单击 Save 按钮后，出现如图12.18所示的 Tables 的窗口。

一旦新数据被创建，用户需要增加表和索引以便数据能够被储存。在用户的数据库中的所有数据都被存储在表中，每个表被定义为一系列的字段，每个字段描述了一种通过指定的数据类型、大小和其他特性而存储的一种数据。表是同数据联系在一起的，例如，在 BIBLIO.MAB 的数据库中，关于每本书出版者的大多数信息被存储在 Publisher 表中，因为一个出版者能够出版许多书、涉及到许多作者，所以关于作者和标题被存储在它各自的表中。

下面我们将学习怎样建立一个新表

(1) 在图12.18的 Tables 窗口中单击 New 命令创建一个新表，这时出现一个增加表的对话框，见图12.19。


Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
828,1080
5892×4836
New Database
bitmaps
hc
icons
include
metafile
odbc32
report
samples
setupkit
vb.cbt
winapi
biblio.mdb
Wdh.mdb
File name:
Save
Cancel
Start
CD Pl...
Project1
Micros...
Prope...
Form1
Data...
New Data...
6:56 PM

Save in: Vb

Save as type: Access (*.mdb)

图12.17　建立新数据库的对话框

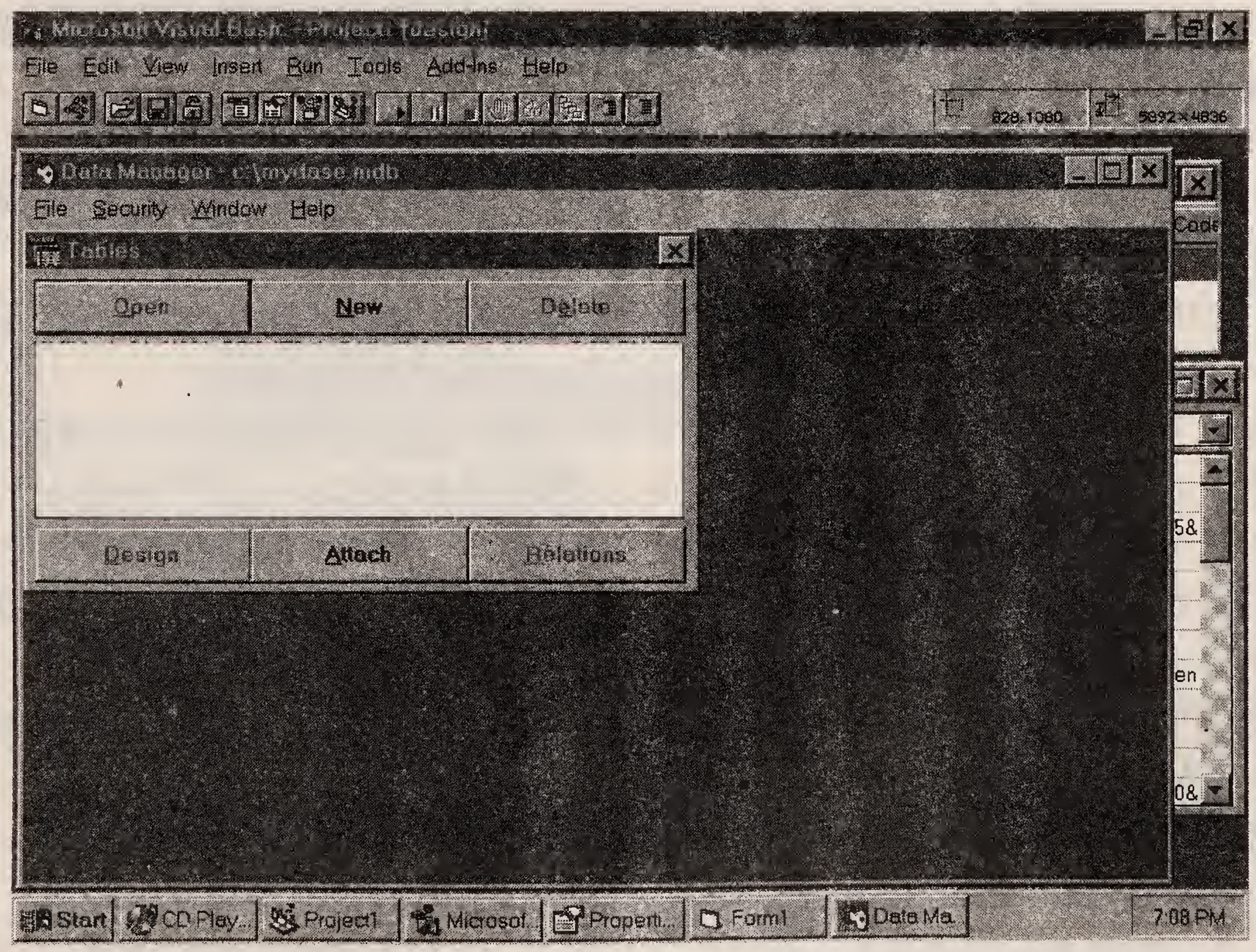

Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
828,1080
5892×4836
Data Manager - c:\mydase.mdb
File Security Window Help
Tables
Open
New
Delete
Design
Attach
Relations
Start
CD Play...
Project1
Microsof...
Properti...
Form1
Data Ma...
7:08 PM

图12.18　Tables 窗口

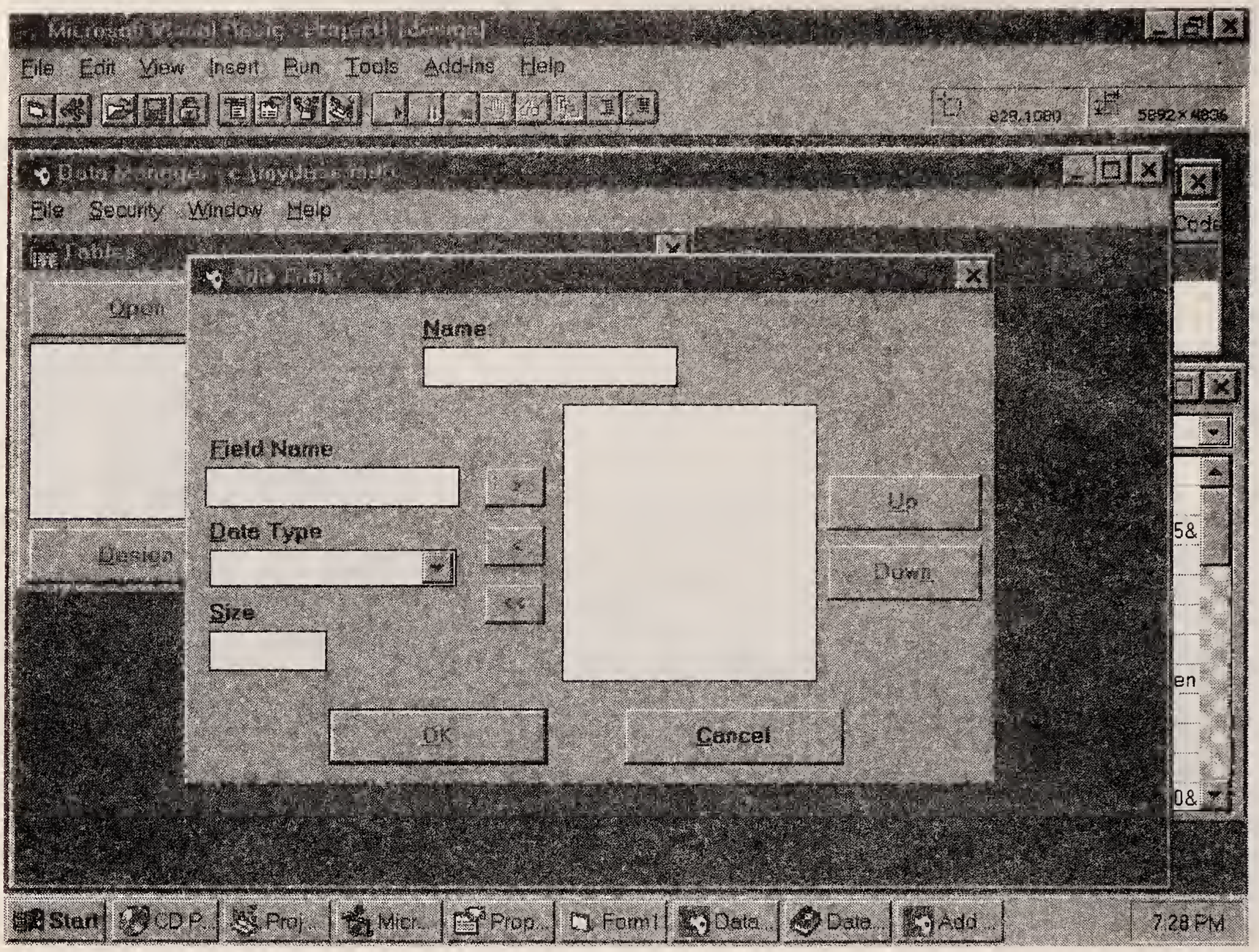

File Edit View Insert Run Tools Add-Ins Help
829,1080
5892 x 4836
File Security Window Help
Tables
Open
Design
Add Table
Name
Field Name
Data Type
Size
>
<
<<
Up
Down
OK
Cancel
Start
CD P...
Proj...
Micr...
Prop...
Form1
Data...
Data...
Add...
7:28 PM

图12. 19 Add Table 对话框

(2) 在 Name 文本框中键入表名。表的命名必须遵循变量名的命名规则，对于外部数据库必须遵循 Access 2. 0所规定的规则，第一个字符应该是字母，而不是数字，例如键入 Student。

(3) 在 FieldName 文本框中键入字段名。对于 Visual Basic 数据库，字段的命名规则同其他的数据库对象的命名规则一样。

(4) 选择字段的数据类型。这决定什么类型的信息被存储在字段中，例如，字符或数字。单击 Datatype 列表框的下箭头打开列表，单击需要的数据类型，见图12. 20。

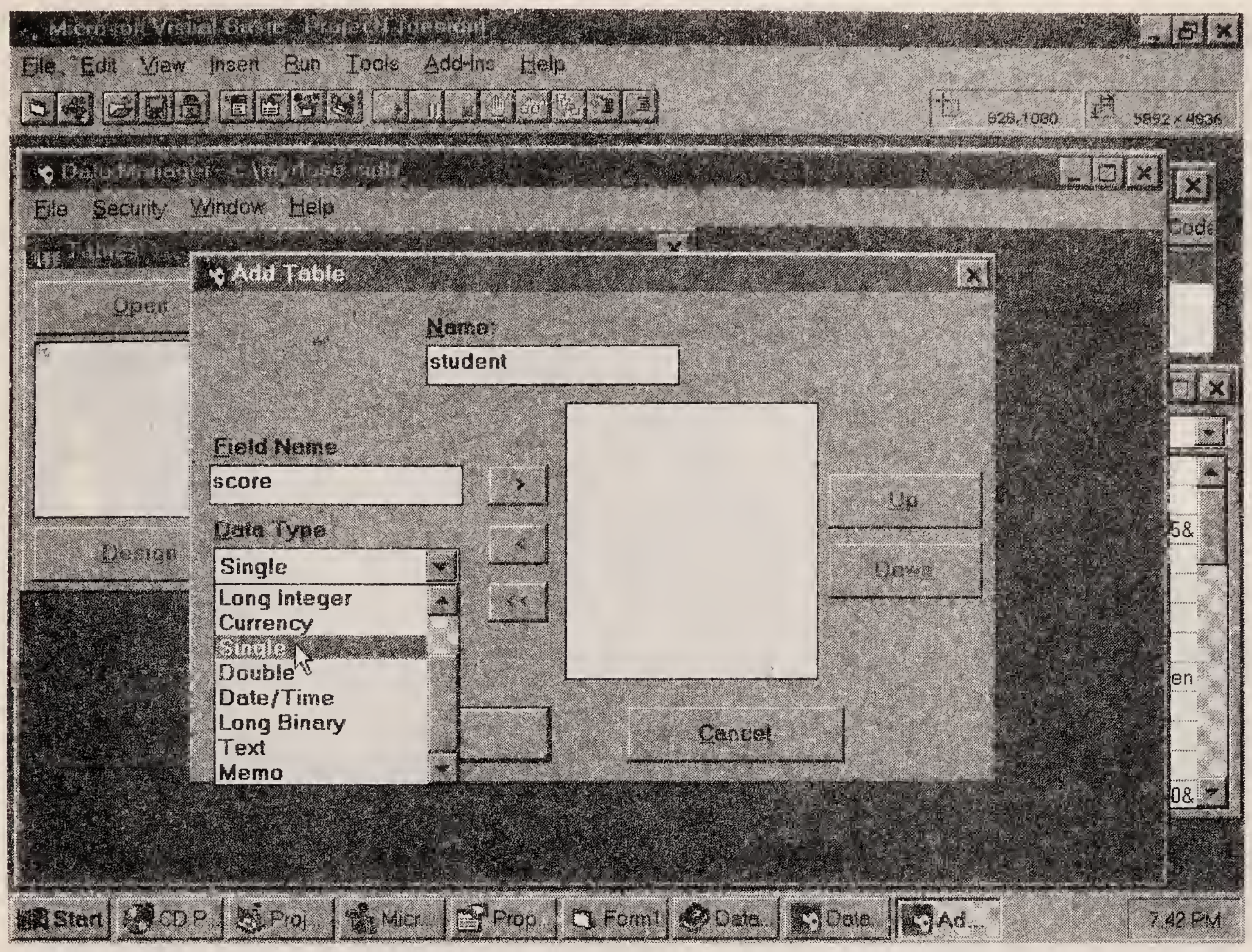
对于大多数的数据类型，数据的大小是自动设置的。但是对于文本字段，应该键入被存储在字段中的最大字符数。用户必须记住它的记录长度被限制在2000个字符以内。

(5) 单击 FileName 文本框旁边的字段增加按钮“>”，增加新字段到字段窗口中，该窗口用于增加新字段到用户的表中，见图12. 21。用户可以看到有一个 Score 字段被增加到字段窗口中。

注意 用户能够增加字段到数据库中，但是在没有删除整个表和重建一个字段的情况下，用户不能对一个字段进行更名、修改或者删除。

用户可以重复(3)—(5)步骤所需要的字段到用户的表中。然后，单击 OK 回到 Tables 窗口，并且窗口内出现新创建的表(Student)。

(6) 在 Tables 窗口单击新建的表使它处于激活状态，此时单击 Design 按钮，它将装载 Table Editor 窗口，见图12. 22。



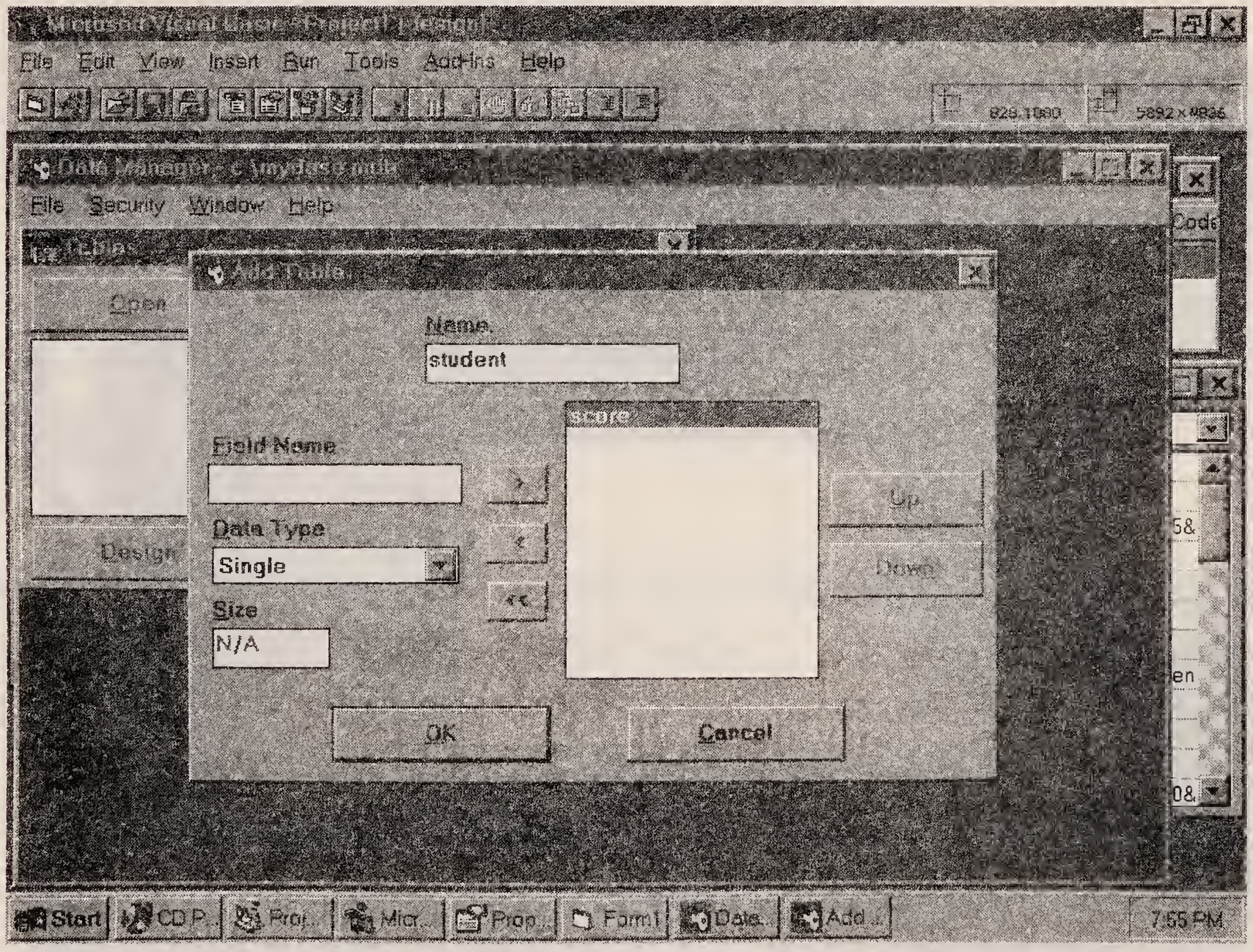
Add Table
File Security Window Help
Open
Design
Long Integer
Currency
Single
Double
Date/Time
Long Binary
Text
Memo
Up
Down
Cancel
Start
7:42 PM

Name: student

Field Name: score

Data Type: Single

图12.20　选取字段的数据类型



Add Table
File Security Window Help
Open
Design
score
Up
Down
OK
Cancel
Start
7:55 PM

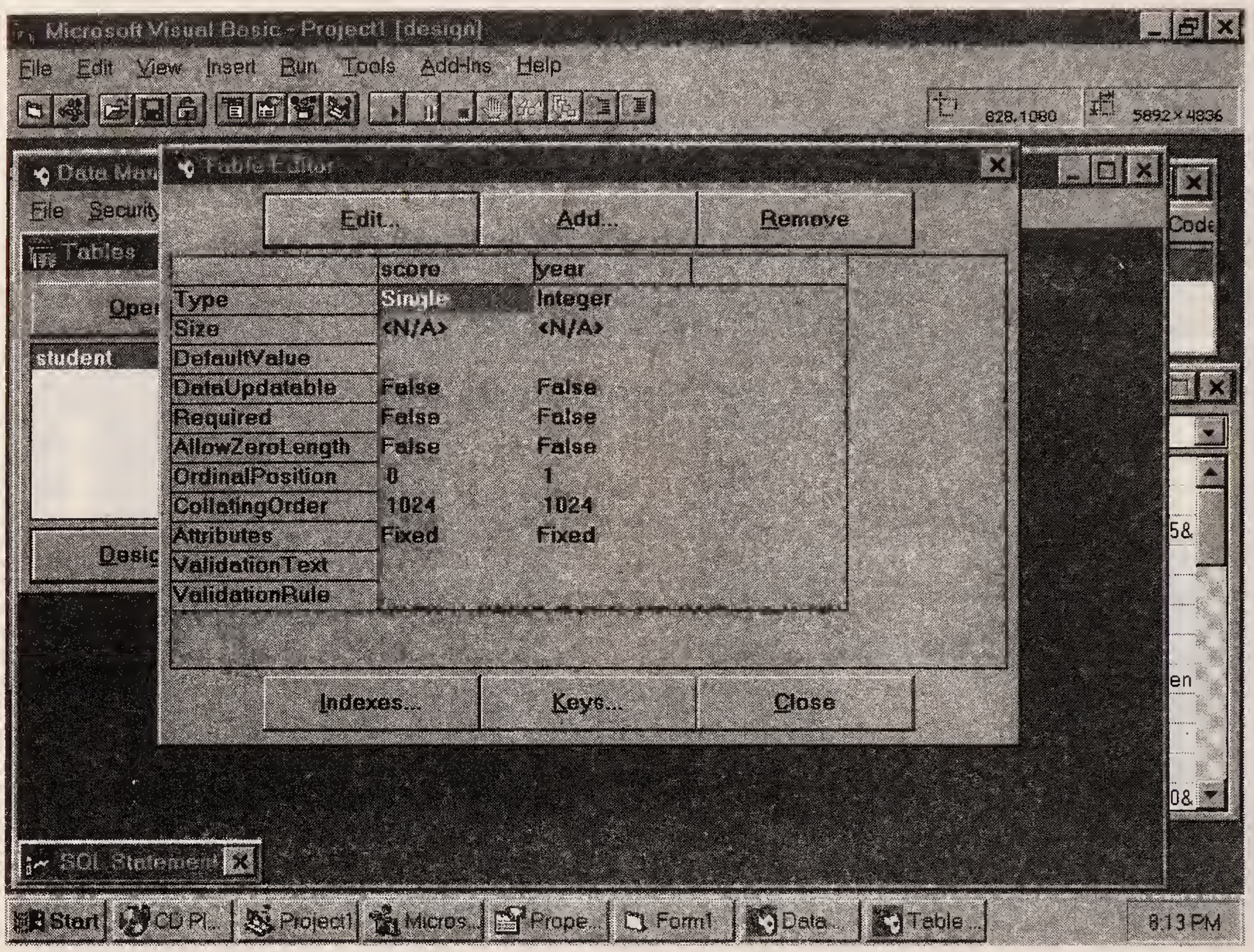
Name: student

Field Name:

Data Type: Single

Size: N/A

图12.21　增加新字段到字段窗口中


Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
828,1080
5892×4836
Data Manager
File Security
Tables
Open
student
Design
Table Editor
Edit...
Add...
Remove
Indexes...
Keys...
Close
SQL Statement
Start
CD Pl...
Project1
Micros...
Prope...
Form1
Data...
Table...
8:13 PM

	score	year
Type	Single	Integer
Size	<N/A>	<N/A>
DefaultValue		
DataUpdatable	False	False
Required	False	False
AllowZeroLength	False	False
OrdinalPosition	0	1
CollatingOrder	1024	1024
Attributes	Fixed	Fixed
ValidationText		
ValidationRule		

图12.22 Table Editor 窗口

（7）单击 Indexes 按钮产生 Indexes 窗口，见图12.23。

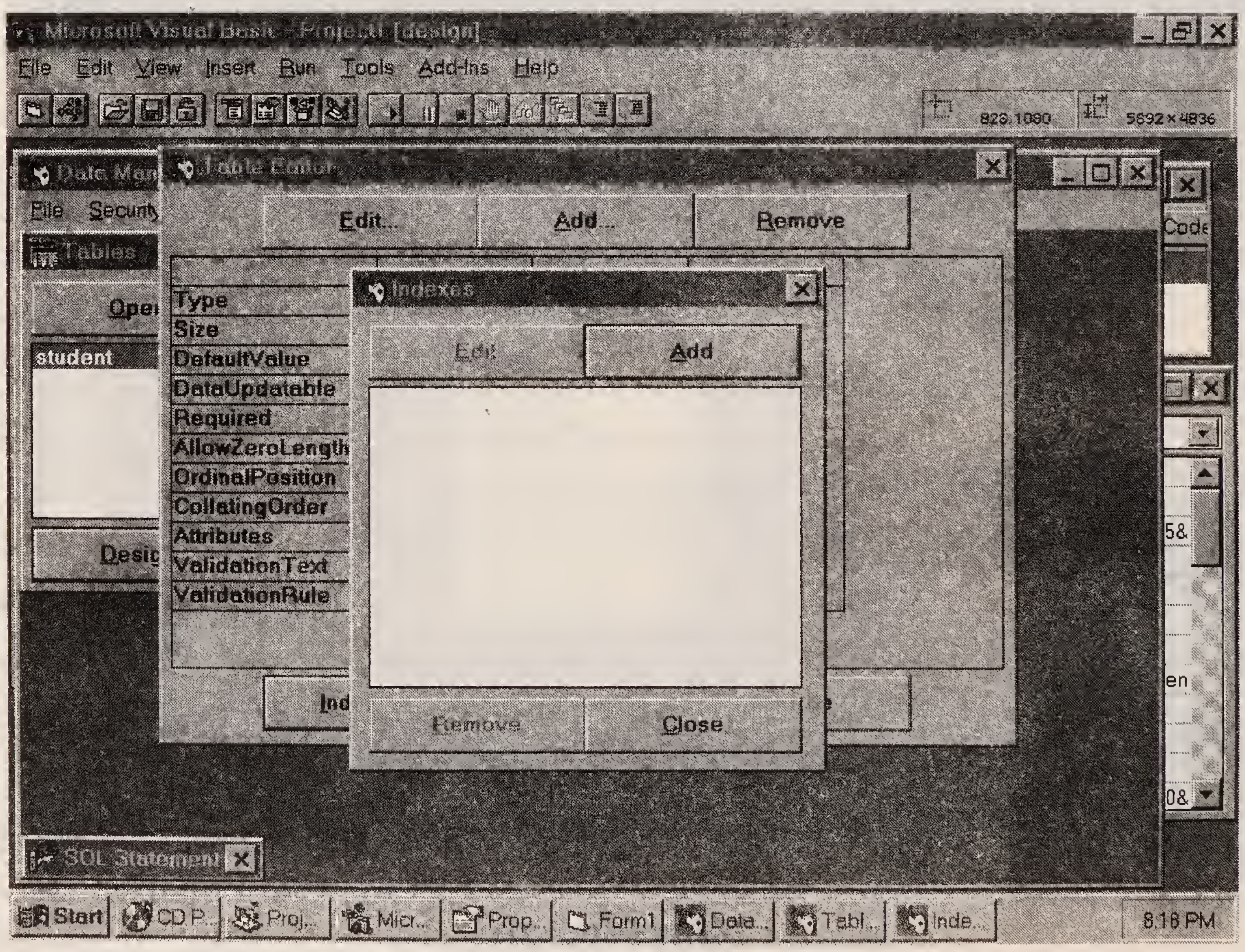

Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
828,1080
5892×4836
Data Manager
File Security
Tables
Open
student
Design
Table Editor
Edit...
Add...
Remove
Type
Size
DefaultValue
DataUpdatable
Required
AllowZeroLength
OrdinalPosition
CollatingOrder
Attributes
ValidationText
ValidationRule
Indexes
Edit
Add
Remove
Close
SQL Statement
Start
CD P...
Proj...
Micr...
Prop...
Form1
Data...
Tabl...
Inde...
8:16 PM

图12.23 Indexes 窗口

(8) 单击 Add 按钮，打开 Add Index 窗口用以用户表所需的索引，见图12.24。

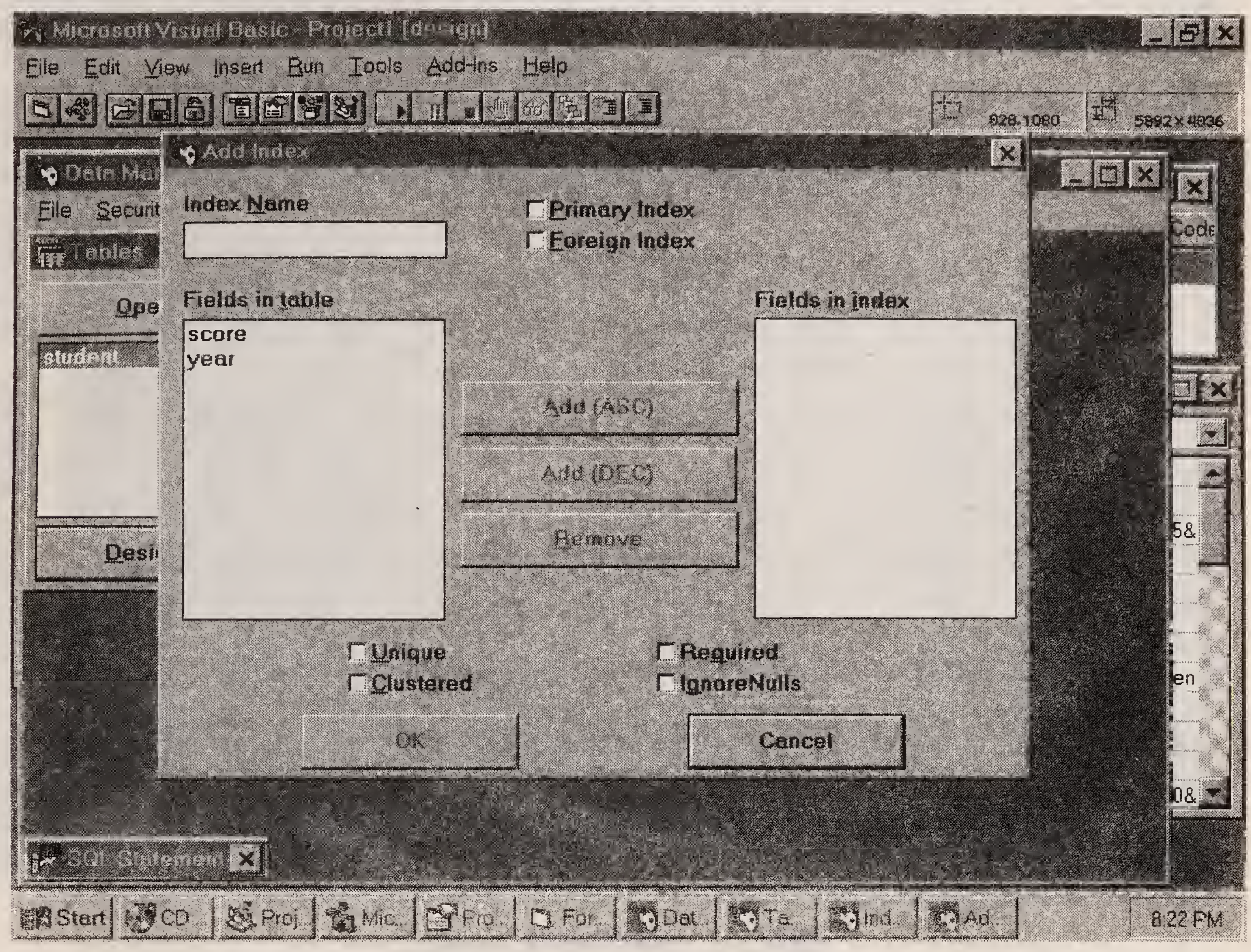

Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
928,1080
5892×4836
Add Index
Data Ma
File Securit
Tables
Ope
student
Desi
Index Name
Primary Index
Foreign Index
Fields in table
score
year
Fields in index
Add (ASC)
Add (DEC)
Remove
Unique
Clustered
Required
IgnoreNulls
OK
Cancel
SQL Statement
Code
Start
CD
Proj
Mic
Fro
For
Dat
Ta
Ind
Ad
8:22 PM

图12.24 Add Index 窗口

(9) 在“Add Index”窗口的 Index Name 文本框内键入索引号，例如 WWW，然后选取一个或多个表中的字段，单击 Add(Asc)或 Add(DEC)按钮为表创建索引(Add(ASC)或 Add(DEC)分别决定是按升序或降序排列)，最后单击 OK 回到“Indexes”窗口，同时表的索引号 WWW 出现在窗口中。

(10) 用户可以重复以前的步骤直到数据库的所有表都被创建。

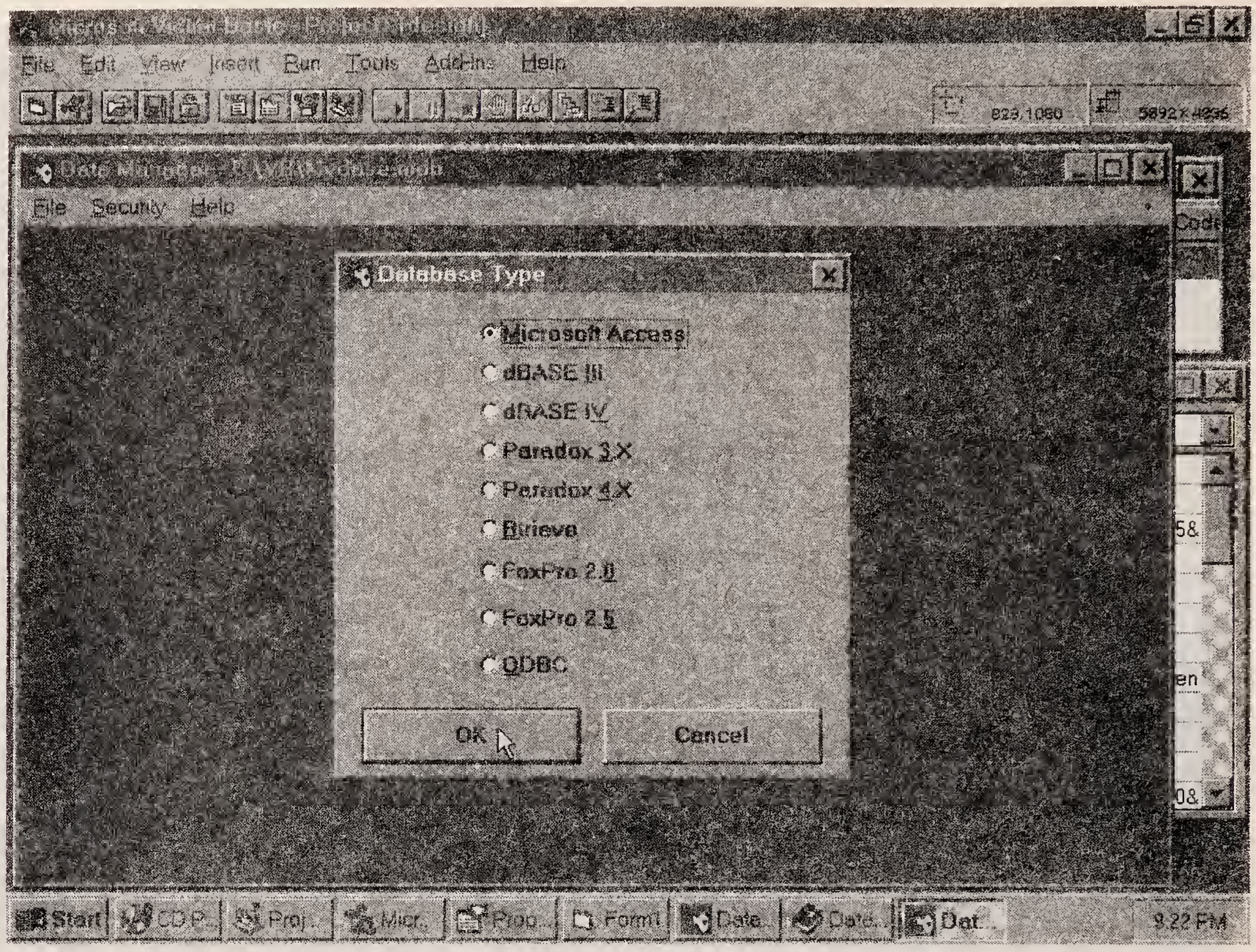
一旦用户的数据库结构被创建，用户能用数据控件编写 Visual Basic 程序填写的数据库，或者用数据管理器打开表直接编辑它。

12.3.2 为数据库添加信息

用户可以利用数据控件通过编写代码来为数据库添加信息，这一点在上一节已有所涉及，此处讲一讲如何用数据管理器向数据库添加信息。

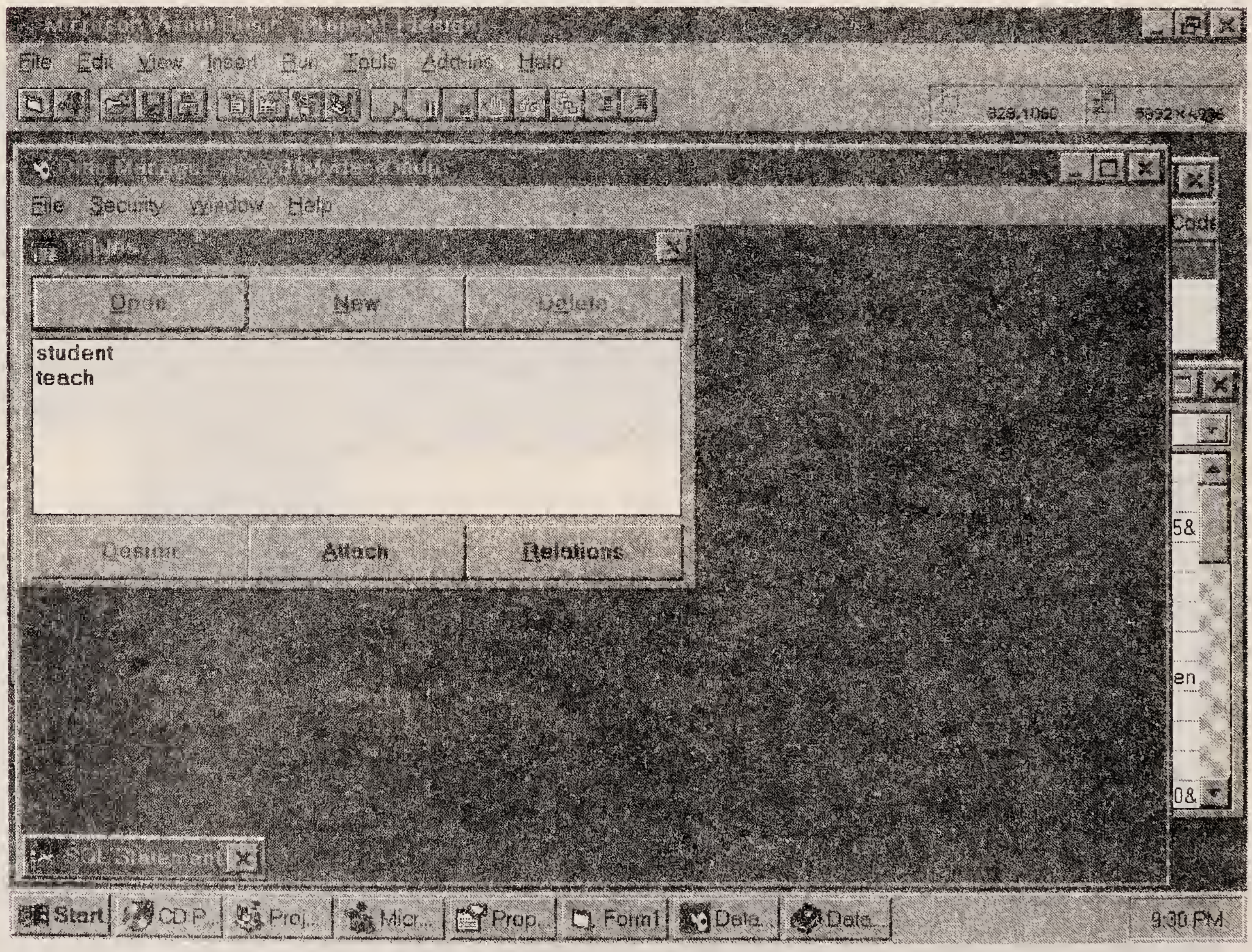
用户可以按以下步骤为数据库添加信息：

(1) 单击数据管理器主屏幕 File 菜单的 Open Database 命令，此时出现 Database Type (数据库类型)窗口，见图12.25。选取所需的数据库类型(此处选取 Microsoft Access)，单击 OK。打开“Open Database”对话框，选取我们刚才所建的 MYDASE.MDB 数据库，单击 Open 打开 MYDASE.MDB 数据库，见图12.26。


File Edit View Insert Run Tools Add-Ins Help
File Security Help
Database Type
OK
Cancel
Start CD P... Proj... Micr... Prop... Form1 Data... Data... Dat...
9:22 PM

● Microsoft Access
○ dBASE III
○ dBASE IV
○ Paradox 3.X
○ Paradox 4.X
○ Btrieve
○ FoxPro 2.0
○ FoxPro 2.5
○ ODBC

图12. 25 DataDase Type 窗口


File Edit View Insert Run Tools Add-Ins Help
File Security Window Help
Open
New
Delete
Design
Attach
Relations
SQL Statement
Start CD P... Proj... Micr... Prop... Form1 Data... Data...
9:30 PM

student
teach

图12. 26 MYDASE. MDB 数据库

（2）由图12.26可以看出，Mybase.mdb 数据库有两个表：Student 和 Teach。单击 Student 表使它处于激活状态，单击 Open 按钮打开此表，见图12.27。

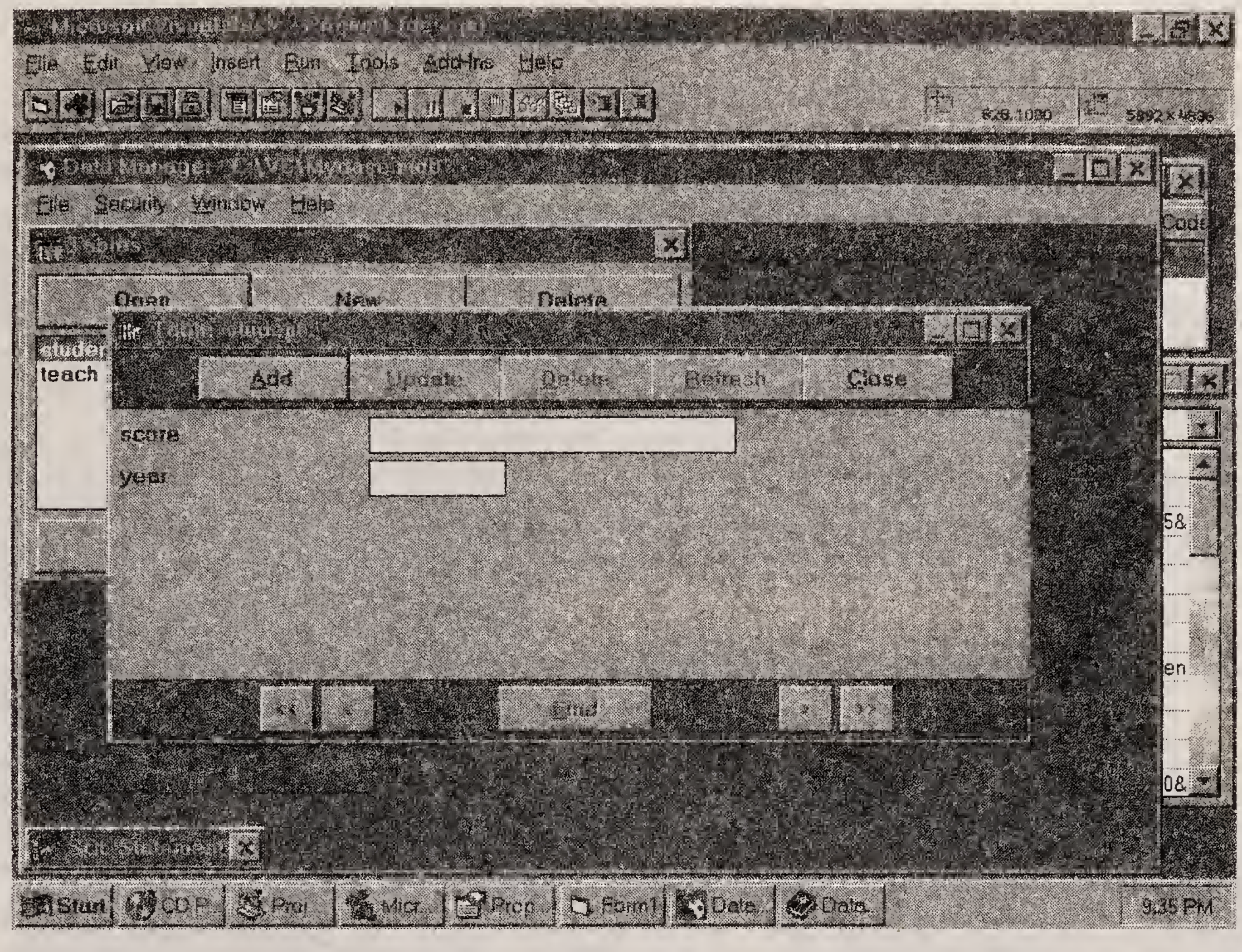
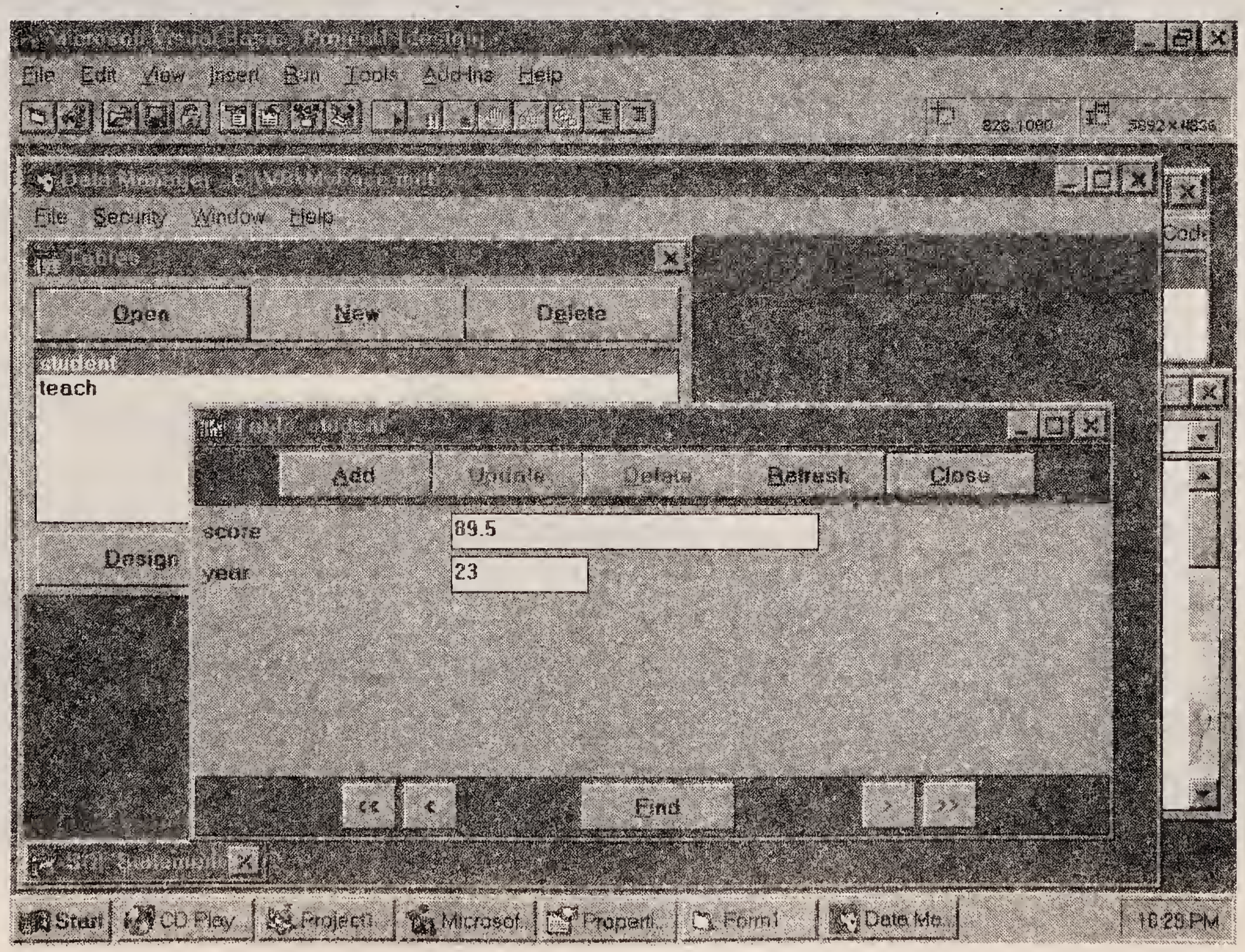

File Edit View Insert Run Tools Add-Ins Help
828,1080
5892 x 4635
Data Manager
File Security Window Help
Tables
Open
New
Delete
student
teach
Add
Update
Delete
Refresh
Close
score
year
<<
<
Find
>
>>
SQL Statement
Start
9:35 PM

图12.27 打开的 Student 表

（3）由图12.27可以看出，Student 表有两个字段：score 和 year。用户单击 Add 按钮后，向每个字段键入信息，例如键入89.5和23。最后单击 Update 按钮即可把信息添加到 MYDASE.MDB 数据库中，见图12.28。


File Edit View Insert Run Tools Add-Ins Help
828,1080
5892 x 4635
Data Manager
File Security Window Help
Tables
Open
New
Delete
student
teach
Design
Add
Update
Delete
Refresh
Close
<<
<
Find
>
>>
SQL Statement
Start
CD Play
Project1
Microsof
Properti
Form1
Data Ma
10:28 PM

score 89.5
year 23

图12.28 添加信息到数据库

(4) 用户可以重复上述步骤向数据库添加更多个记录。

12.3.3 表操作

用户可以通过打开现有数据库来对数据库内各表数据进行各种操作。

例如可以打开 BIBLIO.MDB 数据库，屏幕出现如图12.29所示的“Tables”窗口。

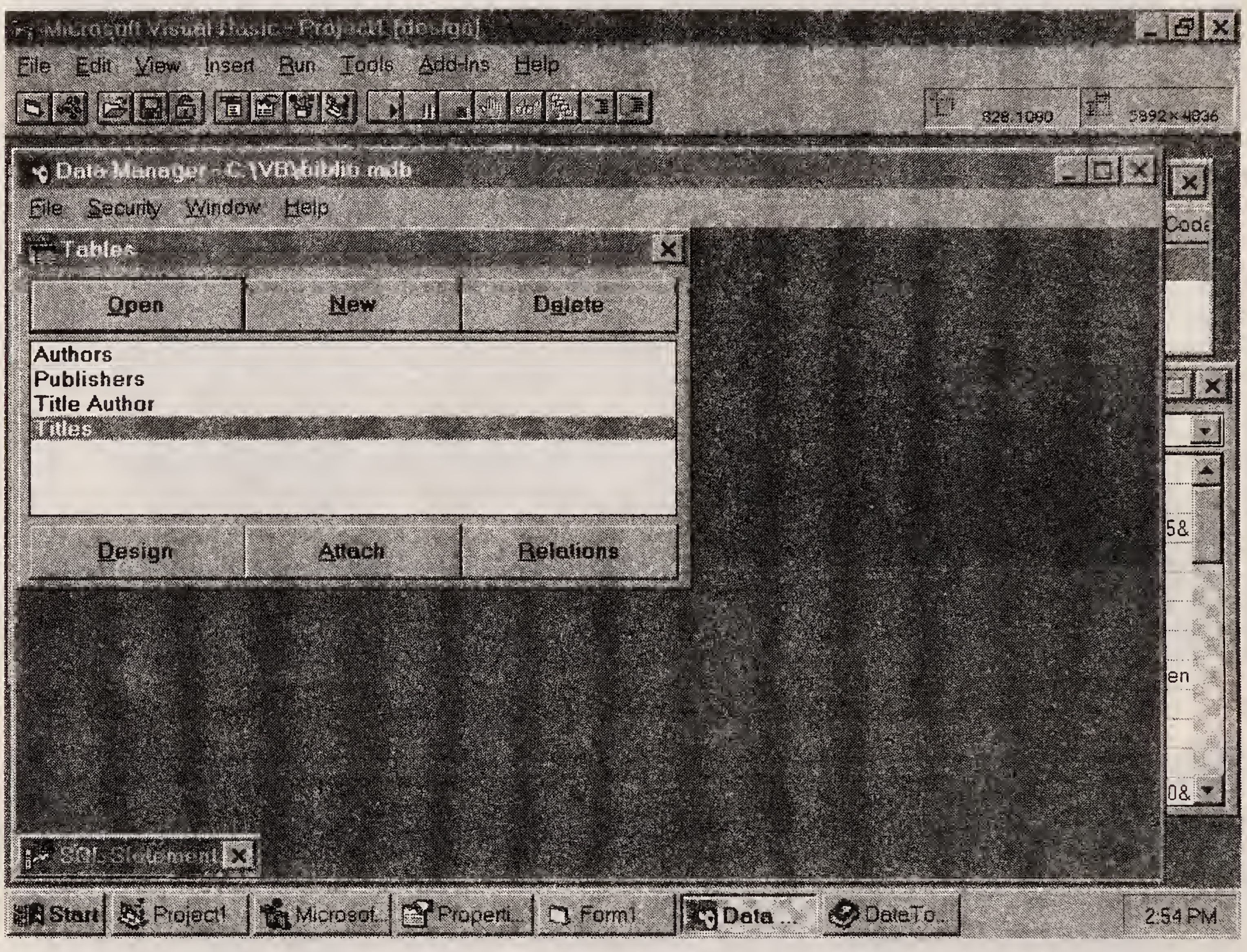

Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
Data Manager - C:\VB\biblio.mdb
File Security Window Help
Tables
Open
New
Delete
Authors
Publishers
Title Author
Titles
Design
Attach
Relations
SQL Statement
Start
Project1
Microsof...
Properti...
Form1
Data ...
DataTo...
2:54 PM

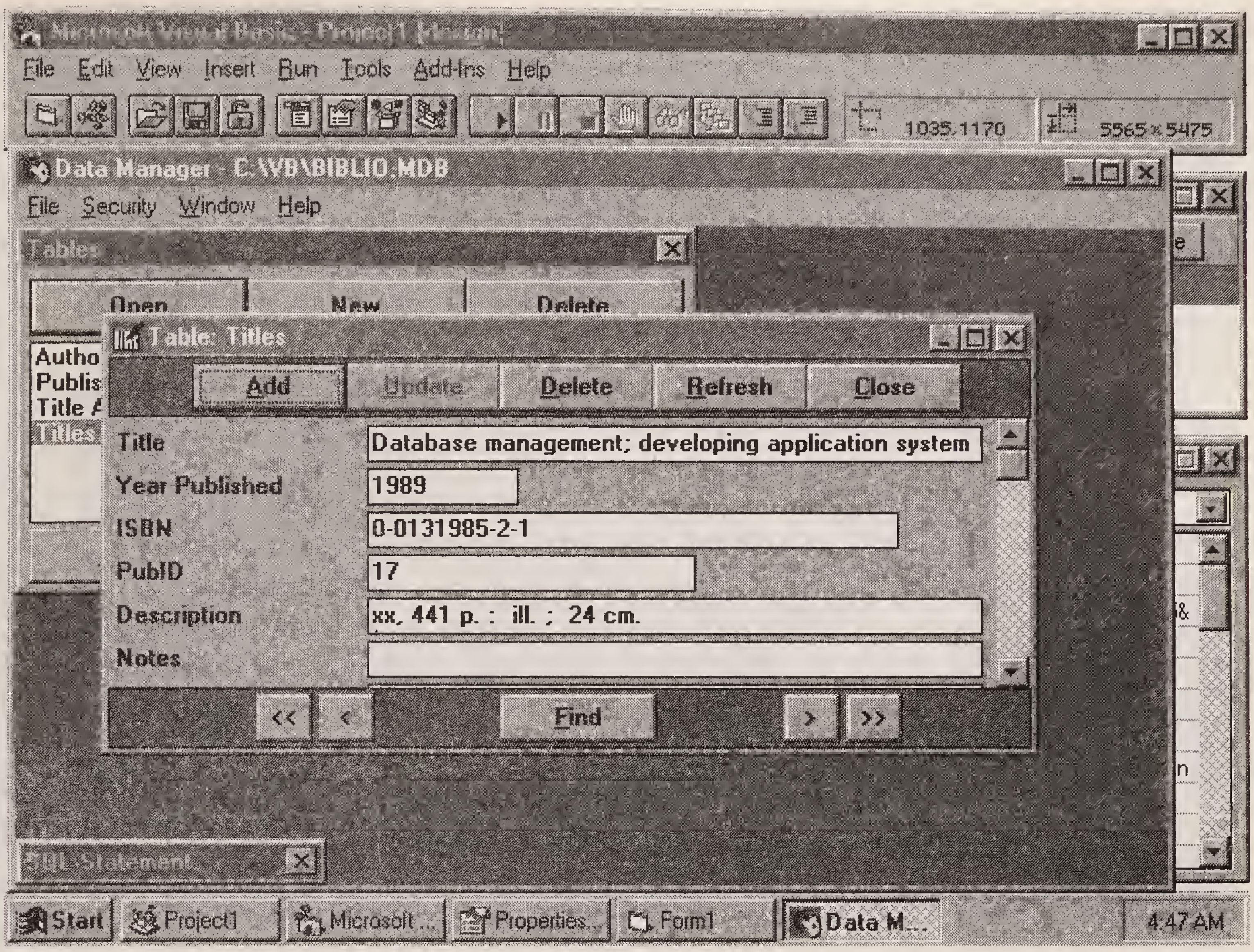
图12.29 BIBLIO.MDB 数据库的 Tables 窗口

从图12.29可以看出 Bibilo.mdb 数据库含有四个表，如果用户想删除一个表，可以先单击该表使之处于激活状态，然后单击 Delete 按钮。单击 Open 按钮可以打开选择的表，例如单击 Title 使之处于激活状态，然后单击 Open 可以打开 Title 表，见图12.30。

现在用户可以通过单击 Find 按钮查找有关记录，也可利用 Find 按钮两边按钮在各记录之间移动。如果用户想删除一个记录，可先用 Find 按钮打开“FindRecord”对话框，然后键入相关的匹配 ，单击 OK，见图12.31。寻找到记录以后再单击 Delete。

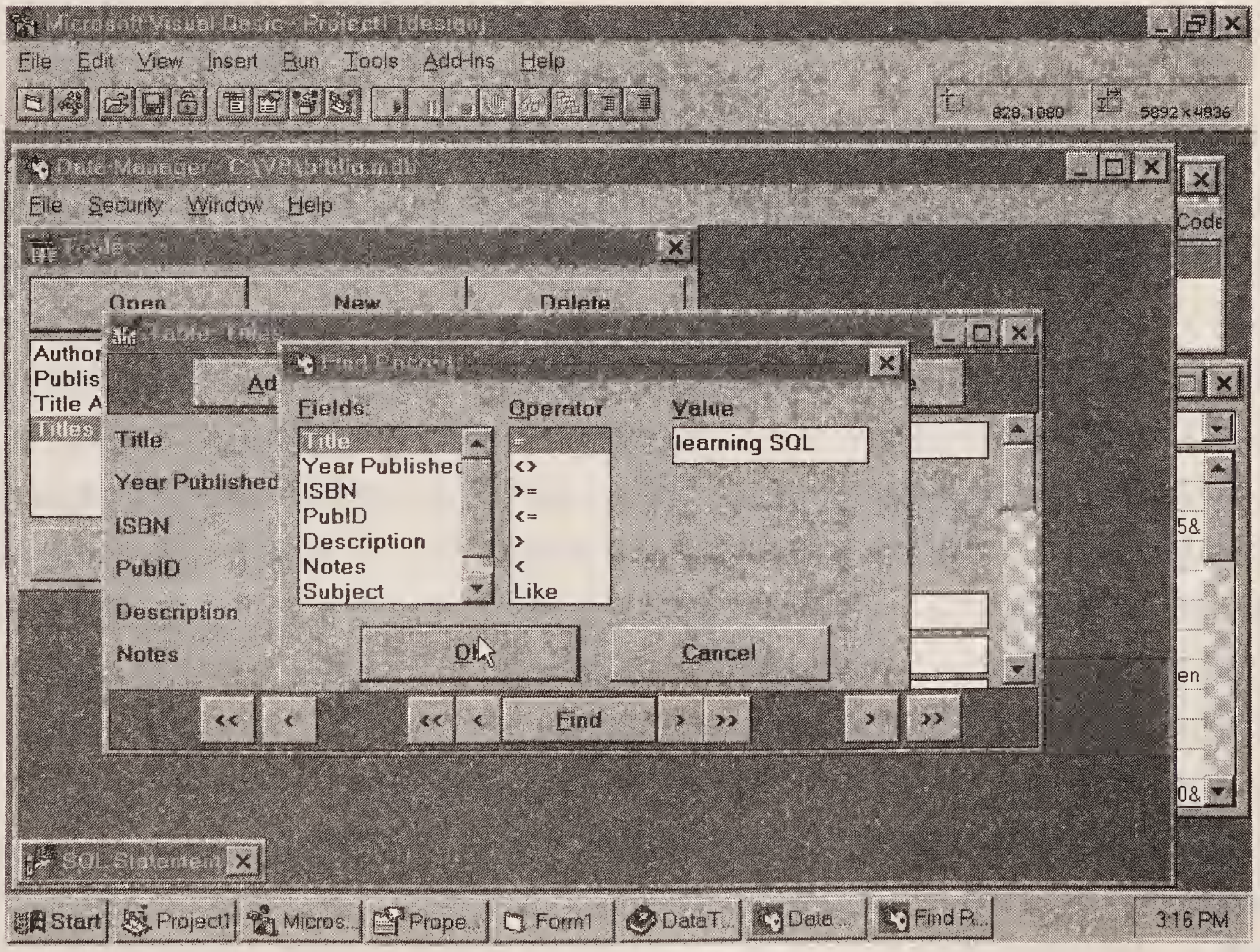
用户可单击“Tables”窗口的 Design 按钮打开“Table Editor”窗口，见图12.32。

在“Table Editor”窗口中，可以通过单击 Edit 按钮打开“Edit Field”窗口对现有的字段进行编辑。用户如果想增加一个新字段到表中，可以单击 Add 按钮打开“Add Field”窗口。用这个窗口，用户可以键入字段名、大小数据类型等方面的内容。


Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
1035,1170
5565 x 5475
Data Manager - C:\VB\BIBLIO.MDB
File Security Window Help
Tables
Open New Delete
Autho
Publis
Title A
Titles
Table: Titles
Add Update Delete Refresh Close
<< < Find > >>
SQL Statement
Start Project1 Microsoft ... Properties... Form1 Data M...
4:47 AM

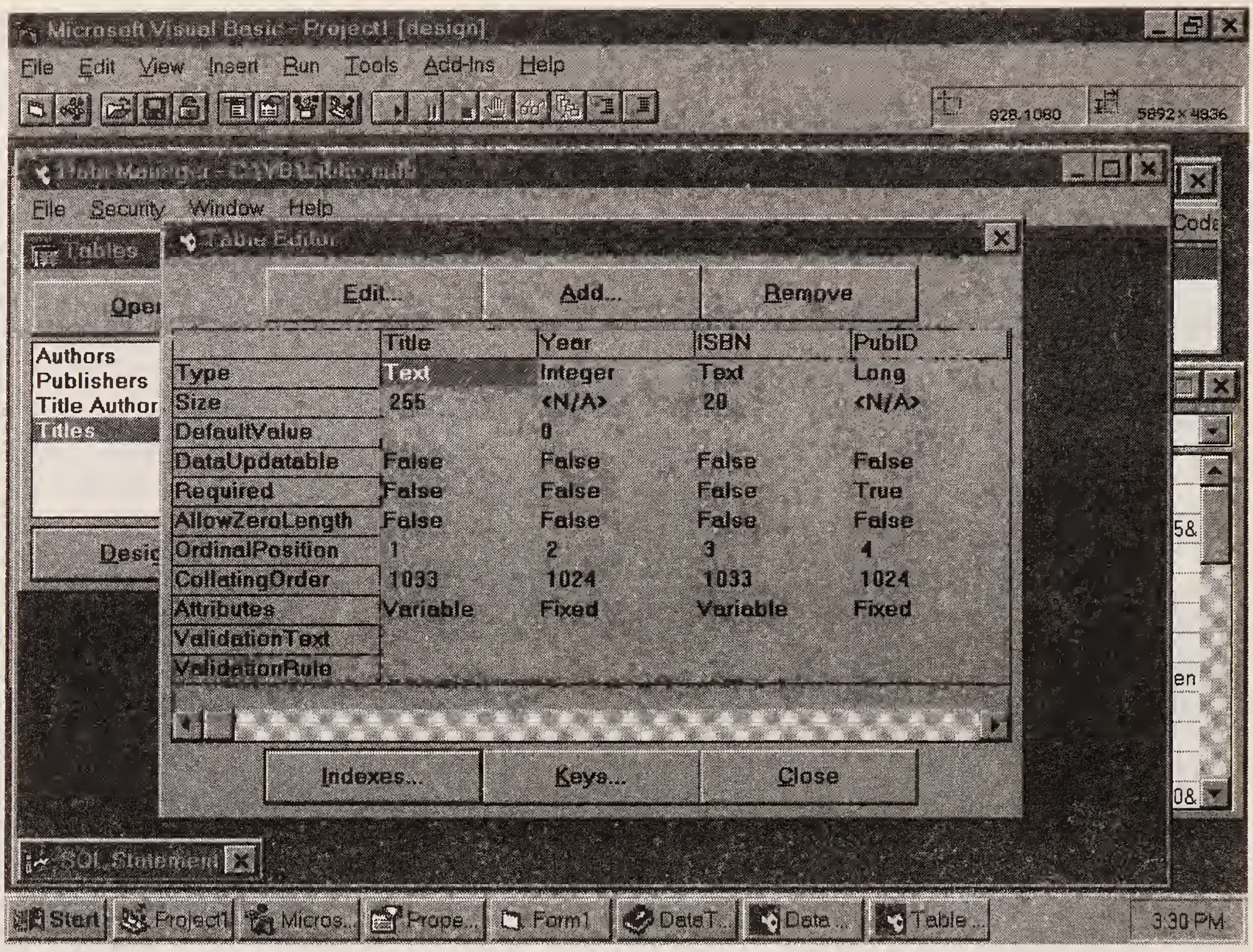
Title: Database management; developing application system
Year Published: 1989
ISBN: 0-0131985-2-1
PubID: 17
Description: xx, 441 p. : ill. ; 24 cm.
Notes:

图12.30 打开的 Title 表


Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
828,1080
5892 x 4836
Data Manager - C:\VB\biblio.mdb
File Security Window Help
Tables
Open New Delete
Author
Publis
Title A
Titles
Table: Titles
Title
Year Published
ISBN
PubID
Description
Notes
Find Record
Fields:
Title
Year Published
ISBN
PubID
Description
Notes
Subject
Operator
=
<>
>=
<=
>
<
Like
Value
OK Cancel
<< < Find > >>
SQL Statement
Start Project1 Micros... Prope... Form1 DataT... Data... Find R...
3:16 PM

Fields: Title
Operator: =
Value: learning SQL

图12.31 查找记录


Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
File Security Window Help
Tables
Table Editor
Edit... Add... Remove
Open
Authors
Publishers
Title Author
Titles
Design
Indexes... Keys... Close
SQL Statement
Start Project1 Micros... Prope... Form1 DataT... Data ... Table...
3:30 PM

	Title	Year	ISBN	PubID
Type	Text	Integer	Text	Long
Size	255	<N/A>	20	<N/A>
DefaultValue		0		
DataUpdatable	False	False	False	False
Required	False	False	False	True
AllowZeroLength	False	False	False	False
OrdinalPosition	1	2	3	4
CollatingOrder	1033	1024	1033	1024
Attributes	Variable	Fixed	Variable	Fixed
ValidationText				
ValidationRule				

图12.32 表编辑窗口

12.4 对象的链接与嵌入(OLE)

OLE 使得基于 Windows 的应用程序之间能够协作，使用 OLE，Visual Basic 应用程序可以显示和控制其他应用程序的数据。用户能够从工具箱中提供的 OLE 控件插入支持 OLE 的对象到 Visual Basic 的应用程序中，用户能够访问的 OLE 对象取决于用户系统所安装的 Windows 应用程序。例如，Excel Worksheets 工作表单，Excel chart 图，Word document 文档等都是 OLE 对象的例子，如果用户的系统上安装了它们，用户就可以访问这些对象。当用户第一次使用 OLE 控件时出现一个 Insert Object(插入对象)对话框，用户能够用这个对话框来选择需访问的 OLE 对象。本节用户将学会怎样利用 OLE 控件和来自几个 Microsoft Office 应用程序的数据创建一个出版商信息系统的程序，同时用户还学会怎样用对象浏览器检查 Windows 应用程序中的对象，最后用户还将学会用 OLE 自动化技术将 Microsoft Excel 的特点引进到用户的工程文件中。

12.4.1 一些 OLE 术语

OLE 对象

对象是一个独立的信息载体。对象可以是正文、电子表格、图形或其他可以由应用程序控制和显示的东西。一个 OLE 控件一次只能包含一个 OLE 对象。

链接对象

在 OLE 对象中包含可与应用程序链接的数据。一个链接对象包含着被创建该对象的应用程序管理的数据，这些数据存储在远离 Visual Basic 应用程序的一个应用程序文件中，所

以，当用户链接一个对象到 Visual Basic 应用程序中时，只是为被链接到应用程序的 OLE 控件中的对象插入一个地址，而不是插入真正的数据。例如，当用户往一个 Visual Basic 应用程序中链接一个图形时，它的数据是存放在另一个文件中，只有数据文件的路径和数据的一个映像存储到 OLE 控件中。其他的应用程序也能够访问被链接的对象，例如，其他用户能够修改你已经链接到一个 Visual Basic 应用程序上的电子表格。

嵌入对象

OLE 对象的数据可以嵌入到应用程序中。当嵌入一个对象时，在应用程序的 OLE 控件中插入的是对象真正的数据。例如，当用户向一个 Visual Basic 应用程序中嵌入一个 Word 文档时，所有的有关数据都会包含在 OLE 控件中。在使用 Visual Basic 应用程序时，用户可以选择被嵌入的对象，有关的应用程序就自动地启动，对象会自动更新，而更新后的数据存在于 OLE 控件中。同链接对象不同，当一个对象嵌入到一个应用程序中之后，其他对象不能访问被嵌入到对象中的数据。

对象或服务器应用程序

是指释放它们对象的 Windows 应用程序，例如上述创建图形对象的绘图程序就是服务器应用程序。

控制器或客户应用程序

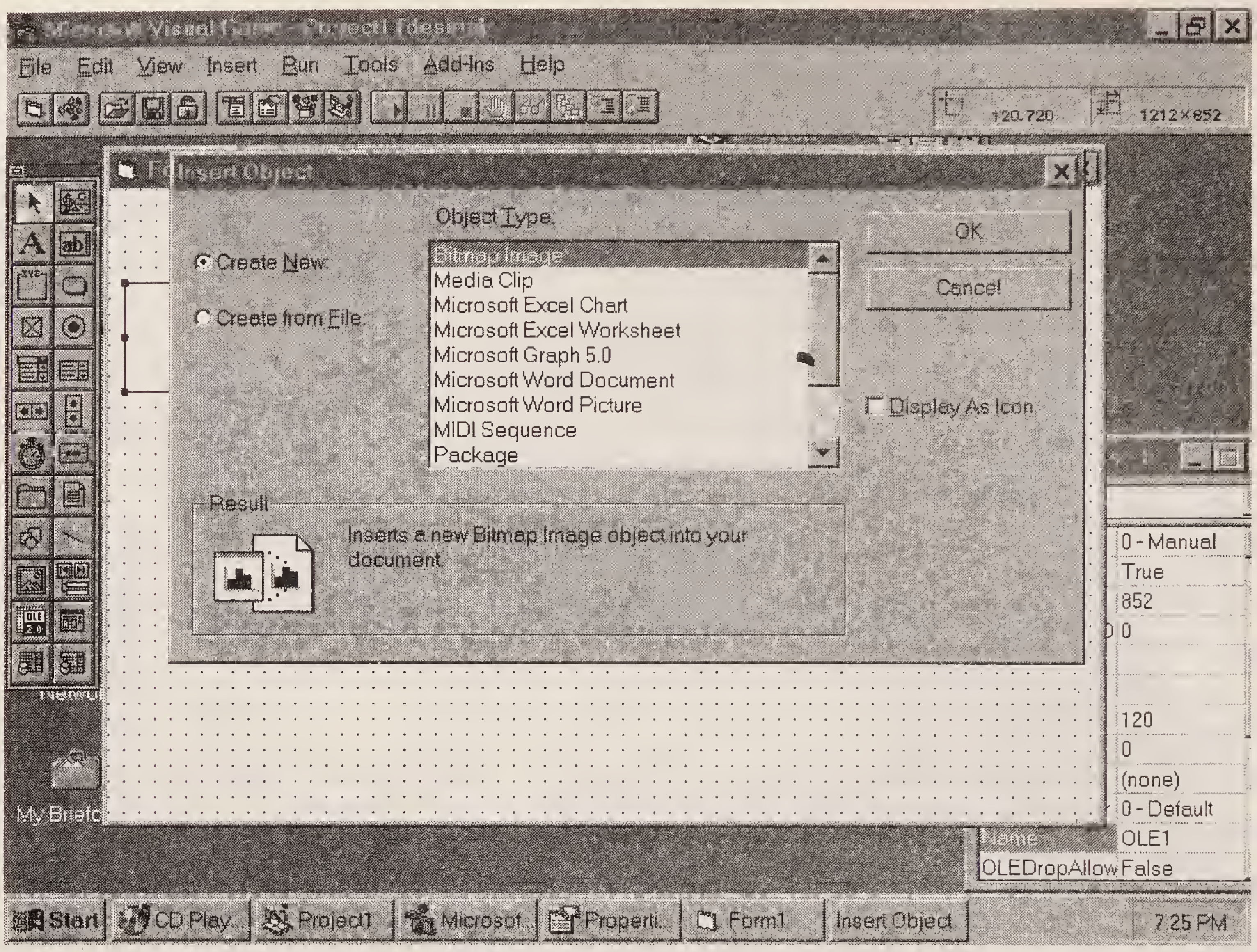
是指包含有另一个应用程序所创建的对象的应用程序。例如，当用户向一个 Visual Basic 应用程序链接或嵌入一个对象时，这个 Visual Basic 应用程序就被称为客户应用程序。

12.4.2 对象的链接与嵌入

前面对链接和嵌入的概念已有所阐述。一般来说链接对象的优点是它占用的空间很少，它的缺点是如果数据的最初始版本丢失或改变，可能会失去链接联系。对于嵌入对象，不必担心数据的丢失，但是它的缺点是对象占去了大量的存储空间。显然，在安全时，链接是最好的选择。但是如果要经常改变源程序，而且想保留数据；就应该嵌入对象。

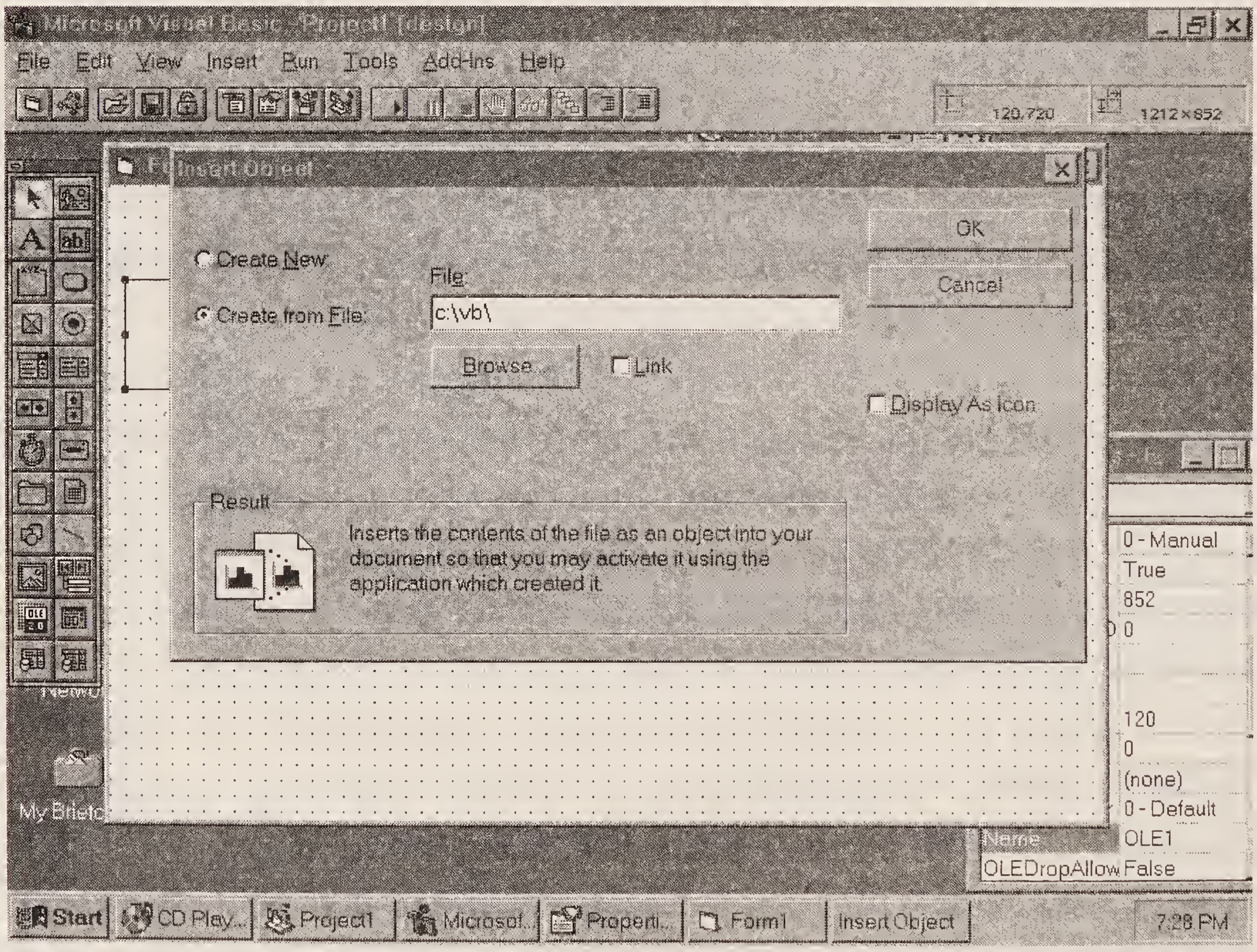
如果用户往一个窗体上添加一个 OLE 2.0控件，就会弹出一个如图12.33所示的对话框，用户拥有的应用程序越多，对话框中可供用户选择的对象也越多。如图12.33所示，在 OLE 2.0控件中相应的对象既可以作为图标显示又可以显示其中的数据，这由 Display As Icon 复选框决定。对话框的最左边两个单选按钮决定用户是链接对象还是嵌入对象。如果用户选中的是 CreateNew（缺省）选项，是让其他应用程序产生一个嵌入对象。如果用户想生成一个链接对象，可以选中 Create from File 选项，这时对话框将变为如图12.34所示，用户可以单击 Browse 按钮打开一个对话框来选取相关的文件，选中文件以后应将图12.34中的 Link 复选框选中，如果用户还想以图标显示，还可以选中 Display As Icon 复选框。

有时候用户还可以通过 Windows 系统的剪贴板来生成链接或嵌入对象。为了达到这一目的，用户必须先把服务器应用程序的有关数据拷贝到剪贴板上。随后，单击插入对话框的 Cancel 按钮后在 OLE 2.0控件上单击鼠标右按钮，这时弹出一菜单，用户单击菜单中的 Paste Special 命令可以打开一个 Paste Special 对话框，见图12.35所示。该对话框左侧也有两个单选按钮决定用户是链接一个对象，还是嵌入一个对象，如果用户想生成一个嵌入式对象，单击 Paste 单选按钮。如果用户想生成一个链接式对象，可以单击 Paste link 单选按钮。


Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
120,720
1212×852
Insert Object
○ Create New:
○ Create from File:
Object Type:
Bitmap Image
Media Clip
Microsoft Excel Chart
Microsoft Excel Worksheet
Microsoft Graph 5.0
Microsoft Word Document
Microsoft Word Picture
MIDI Sequence
Package
OK
Cancel
☐ Display As Icon
Result
0 - Manual
True
852
0
120
0
(none)
0 - Default
Name OLE1
OLEDropAllow False
Start
CD Play...
Project1
Microsof...
Properti...
Form1
Insert Object
7:25 PM

Inserts a new Bitmap Image object into your document.

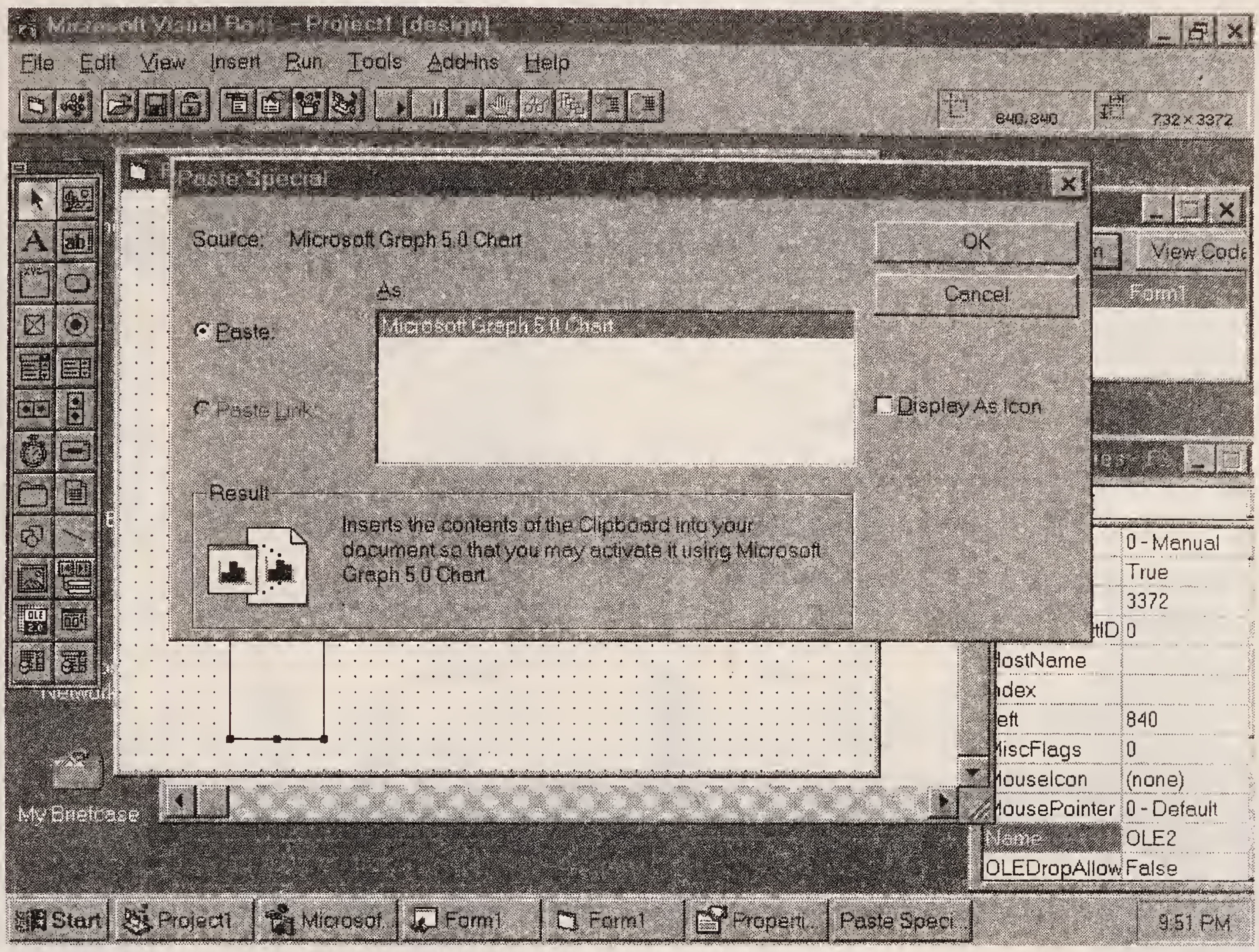
图12.33　OLE 控件的插入对象对话框


Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
120,720
1212×852
Insert Object
○ Create New:
○ Create from File:
File:
Browse...
☐ Link
OK
Cancel
☐ Display As Icon
Result
0 - Manual
True
852
0
120
0
(none)
0 - Default
Name OLE1
OLEDropAllow False
Start
CD Play...
Project1
Microsof...
Properti...
Form1
Insert Object
7:28 PM

File: c:\vb\

Inserts the contents of the file as an object into your document so that you may activate it using the application which created it.

图12.34　OLE 控件的链接对象对话框


Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
840,840
732×3372
Paste Special
OK
Cancel
As:
Microsoft Graph 5.0 Chart
Paste:
Paste Link:
Display As Icon
Result
View Code
Form1
My Briefcase
Start
Project1
Microsof...
Form1
Form1
Properti...
Paste Speci...
9:51 PM

Source:　Microsoft Graph 5.0 Chart

Inserts the contents of the Clipboard into your document so that you may activate it using Microsoft Graph 5.0 Chart.

	0 - Manual
	True
	3372
...ID	0
...ostName	
...dex	
...eft	840
...iscFlags	0
...ouseIcon	(none)
...ousePointer	0 - Default
Name	OLE2
OLEDropAllow	False

图12.35　Paste Special 对话框

在下面的例子中，用户将用一个数据控件访问 BIBLIO.MDB 数据库有关出版商信息，用几个标签和文本框显示每个出版商的信息。另外用户还增加两个 OLE 控件到窗体中，一个链接来自 Excel Worksheet 的数据用来表示出版商在国内外每月的销售额，另一个链接一个 Excel chart 用来表示国内和国外销售量的比较。

例：建立出版商信息管理系统

（1）如果用户已经运行了 Visual Basic 程序，单击 File 菜单的 New Project 命令建立一个新的工程项目文件。

（2）增加标签、文本框和一个数据控件到窗体中，设置它们的属性如下表，建立图12.36所示的初始化界面。

对象	属性	设置
窗体	Caption	Publisherfrm
标签1	Caption	Publisher information system
	Font	Times New Roman，24Point
数据控件	Caption	Publisher
	Connect	Access
	DatabaseName	C:\VB\BIBLIO.MDB
	Readonly	True
	RecordSource	Publishers
文本框1	DataSource	Data1
	DataField	PubID
	Text	空白
标签2	Caption	PubID

（续表）

对象	属性	设置
文本框2	DataSource	Data1
	DataField	Name
	Text	空白
标签3	Caption	Name
文本框3	DataSource	Data1
	DataField	Company Name
	Text	空白
标签4	Caption	Company Name
文本框4	DataSource	Data1
	DataField	Address
	Text	空白
标签5	Caption	Address
命令按钮	Caption	Quit

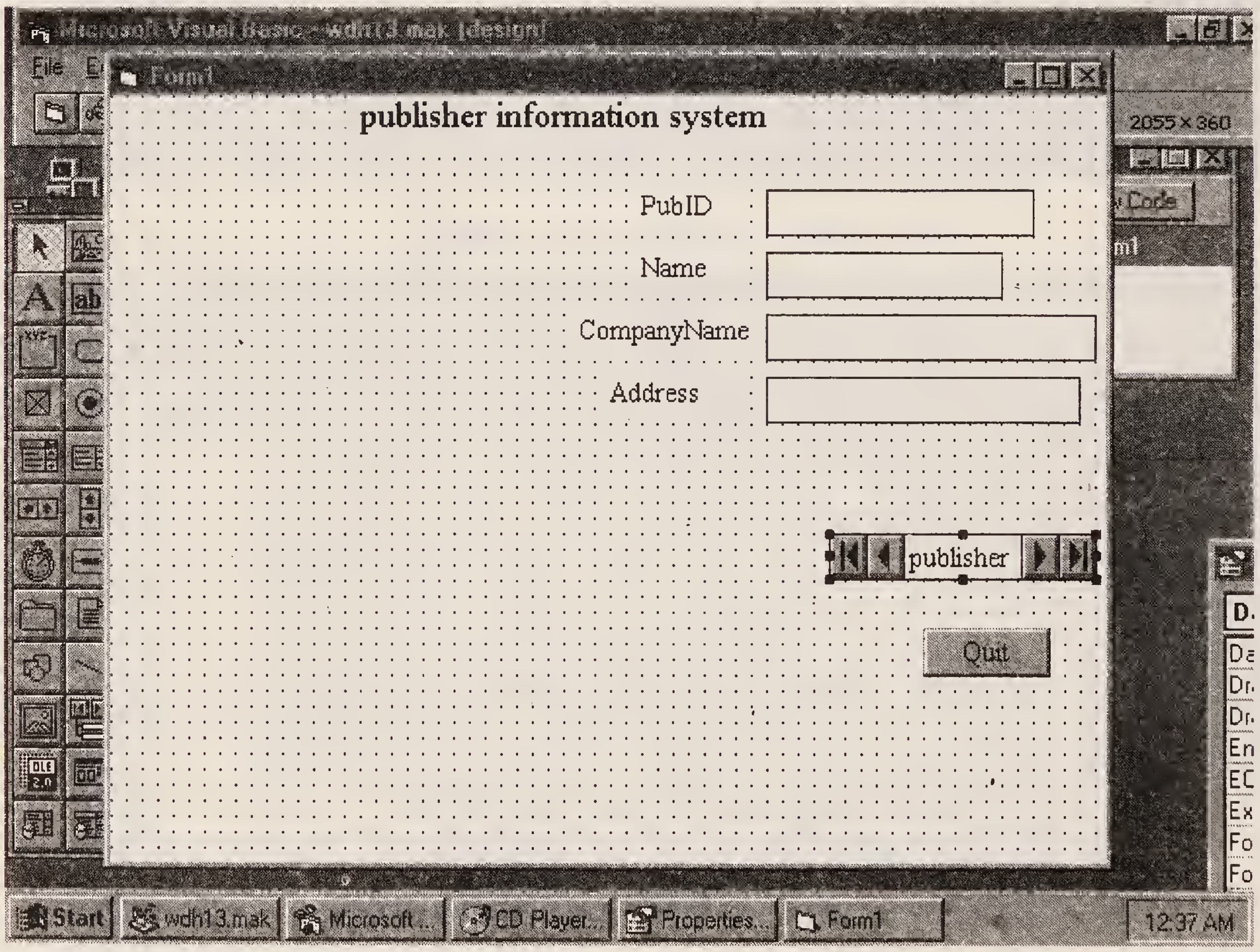
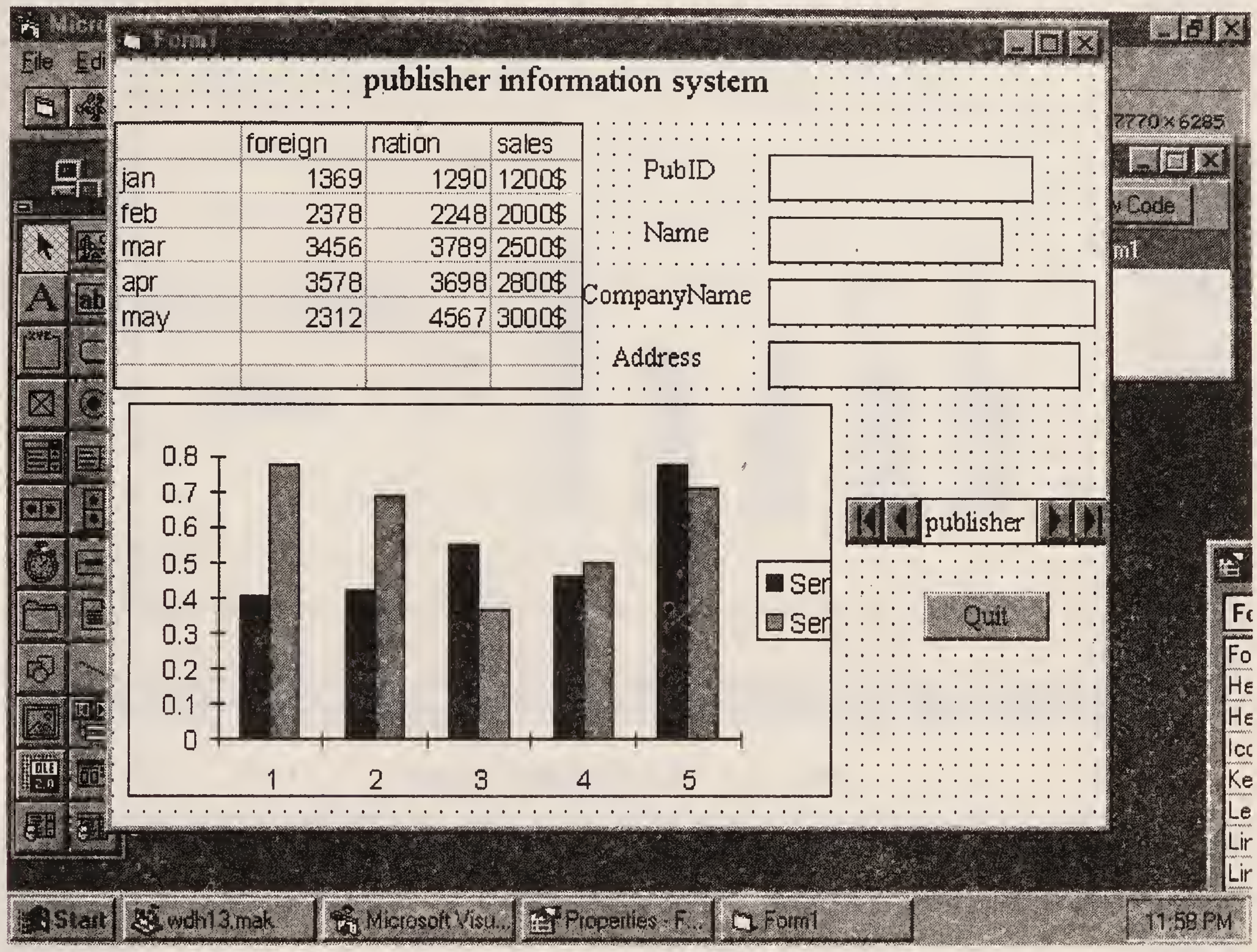

Microsoft Visual Basic - wdh13.mak [design]
File
Form1
publisher information system
PubID
Name
CompanyName
Address
publisher
Quit
2055 x 360
Code
Start
wdh13.mak
Microsoft...
CD Player...
Properties...
Form1
12:37 AM

图12.36　初始化界面

(3) 插入两个对象。单击 OLE 控件，在窗体的左上部拖放出一个方框。当用户释放鼠标时一个 Insert Object（插入对象）对话框出现，见图12.33。这个对话框内有一个用户能够链接或嵌入到他的应用程序的对象列表。单击 Create from File 单选钮，这时出现如图12.34的对话框。为创建一个链接对象到用户的应用程序中，选择 Link 复选框。这意味着用户将选择的文件是一个链接对象，这个文件将继续存在于 Visual Basic 应用程序之外，当用户运行时，该文件的变化将会影响用户的应用程序。为了选择一个文件，用户可单击 Browse 按钮寻找用户想链接到应用程序上的文件。例如，用户可以在 Browse 对话框中选择 C:\MSOFFICE\EXCEL\BOOK2.XLS 文件。在插入对话框中单击 OK 就在用户程序和 book.XLS 文件之间

建立起链接。随后一个 Excel Worksheet 表格出现在 OLE 控件上。这个 worksheet 表明出版商在国内外销售数量和收入。按照同样的方式用户可以链接一个 Excel Chart 对象到用户的应用程序中，该文件是 C:\MSOFFICE\XLSTART\XLSGALRY。到此为止用户的界面应如图12.37所示。


Form1
publisher information system
PubID
Name
CompanyName
Address
publisher
Quit
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
1
2
3
4
5
Ser
Ser
File
Edi
7770×6285
Code
Fo
He
He
Icc
Ke
Le
Lir
Lir
Start
wdh13.mak
Microsoft Visu...
Properties - F...
Form1
11:58 PM

	foreign	nation	sales
jan	1369	1290	1200$
feb	2378	2248	2000$
mar	3456	3789	2500$
apr	3578	3698	2800$
may	2312	4567	3000$

图12.37 链接对象以后的界面

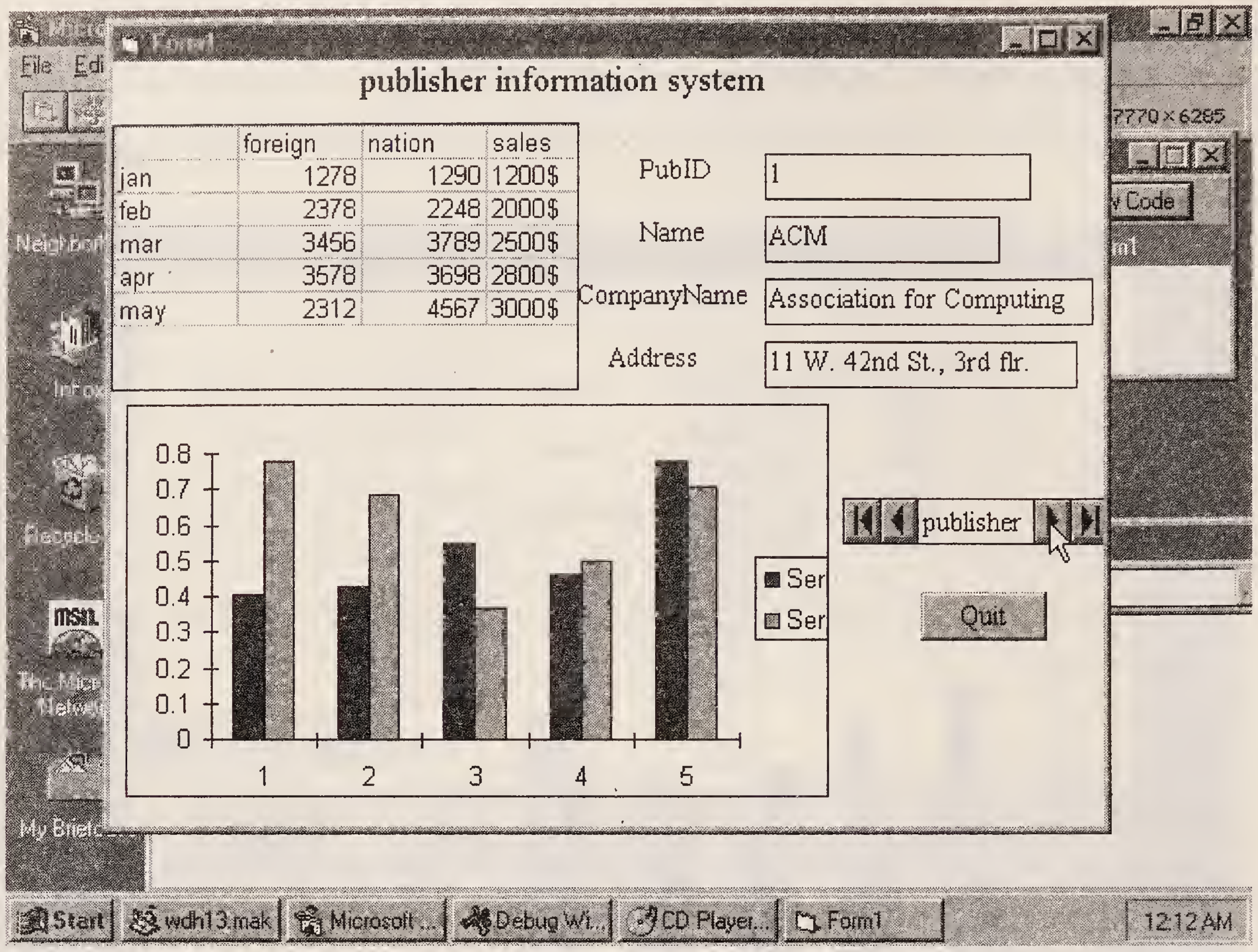
（4）编写代码。双击窗体的空白处打开代码窗口，键入以下代码：

```
Sub Form _ Load()
     OLE1.Update
     OLE2.Update
End Sub
Sub _ Command1 _ Click()
     End
End Sub
```

这些语句用 Update 方法将装载被链接的 Excel 文件的任何变化到用户的应用程序中。End 语句用来退出用户的应用程序。

（5）运行程序。按 F5键或者单击工具条上的 Start 按钮运行程序。这时文本框显示出版商的有关信息，单击数据控件两边的箭头可浏览出各个出版商的有关信息，见图12.38。

由于各个出版商的经营情况不同，所以用户可以对窗体左上部的 Worksheet 单元中的数据进行修改。双击第五个月的国外销售量2312，Visual Basic 将运行 Excel。因为用户插入到应用程序的对象是被链接到 Worksheet，所以 Visual Basic 允许用户对原始的 Worksheet 做一些改变。把光标移到第五个月的国外销售量，键入3789后按回车键，最后单击 File 菜单的 Exit 退出 Excel，这时会出现一个对话框询问用户是否存储对文件的变化，单击 Yes 保存变化。这样原始的 Worksheet 被更新，因为采取的是链接方式，所以这一变化将影响到用户的应用程序，从图12.39可以看出五月份的国外销售量变化为3789。


Form1
File
Edi
7770×6285
Code
m1
Neighbor
Inbox
Recycle
msn.
The Micro Network
My Briefc
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
1
2
3
4
5
■ Ser
▨ Ser
publisher
Quit
Start
wdh13.mak
Microsoft ...
Debug Wi...
CD Player...
Form1
12:12 AM

publisher information system

	foreign	nation	sales
jan	1278	1290	1200$
feb	2378	2248	2000$
mar	3456	3789	2500$
apr	3578	3698	2800$
may	2312	4567	3000$

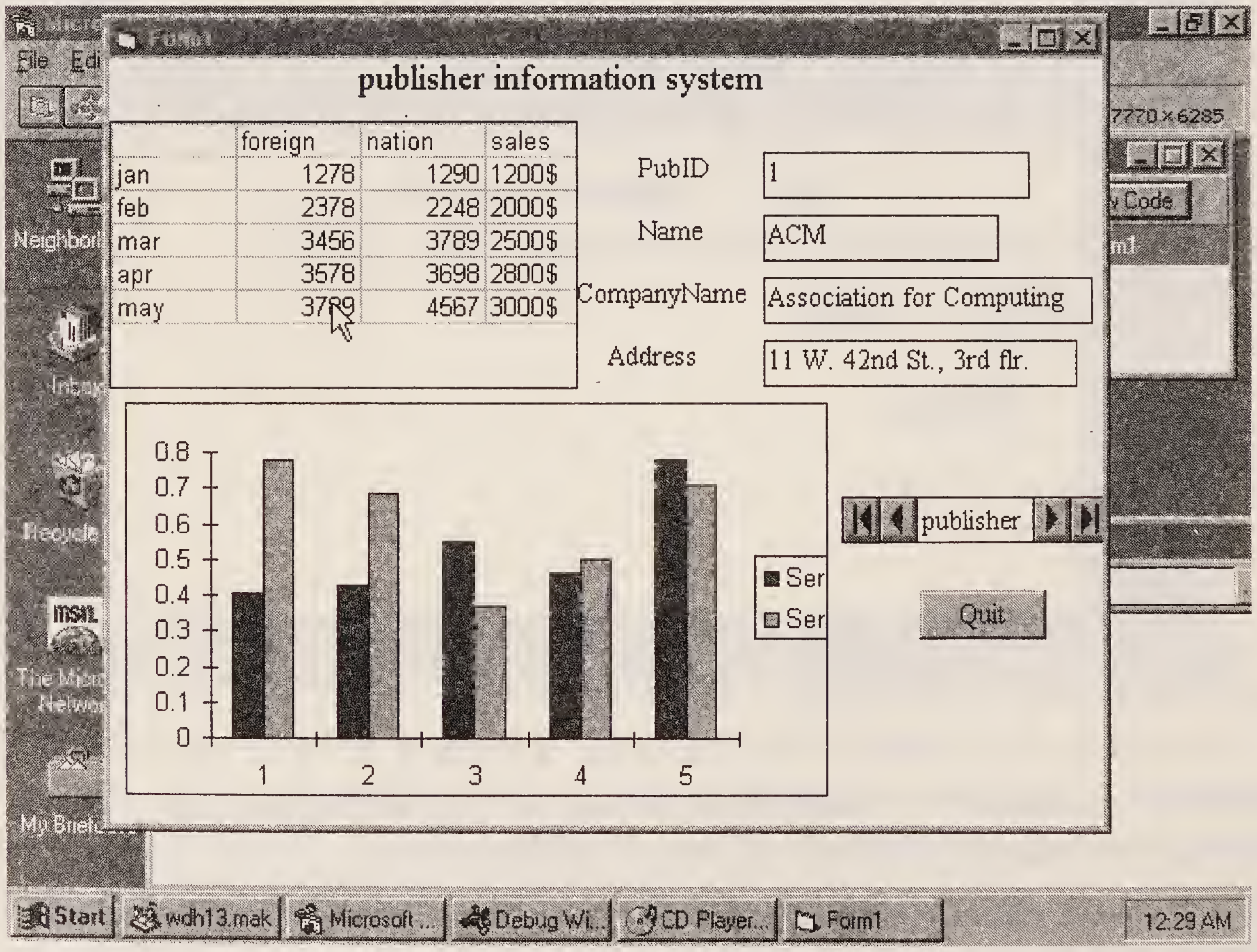
PubID　1

Name　ACM

CompanyName　Association for Computing

Address　11 W. 42nd St., 3rd flr.

图12.38　显示出版商有关信息


Form1
File
Edi
7770×6285
Code
m1
Neighbor
Inbox
Recycle
msn.
The Micro Network
My Briefc
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
1
2
3
4
5
■ Ser
▨ Ser
publisher
Quit
Start
wdh13.mak
Microsoft ...
Debug Wi...
CD Player...
Form1
12:29 AM

publisher information system

	foreign	nation	sales
jan	1278	1290	1200$
feb	2378	2248	2000$
mar	3456	3789	2500$
apr	3578	3698	2800$
may	3789	4567	3000$

PubID　1

Name　ACM

CompanyName　Association for Computing

Address　11 W. 42nd St., 3rd flr.

图12.39　被修改后的信息

（6）单击 Quit 退出程序。

12.4.3 OLE 自动化(Automation)

在前面的部分,用户学会了怎样把基于 Windows 的应用程序的功能引入到用户的 Visual Basic 应用程序中。在这部分中,用户将学会使用被称为 OLE 自动化的技术来引入基于 Windows 应用程序的功能性到用户的代码中。完全支持 OLE 自动化的 Windows 应用程序能够得到或释放它们应用程序的特性,就象带有相关属性和方法的对象集一样。OLE 自动化是一种工业化标准技术,允许两个应用程序之间相互联系。当一个应用程序支持 OLE 自动化,它释放的对象能够被 Visual Basic 访问。通过调用有关对象的方法或者得到和设置对象的属性,Visual Basic 能够操纵这些对象。例如,如果用户创建一个名为 MYOBJ 的自动化对象,用户能够编写以下代码操作这个对象:

```
MYOBJ.Insert "Hello.world"                          'Place text
MYOBJ.Bold=True                                     'Format text
MYOBJ.SaveAs  "C:\WORDPROC\DOCS\TEST.DOC"           'Save the object
```

用下面的函数用户能够访问一个 OLE 自动化对象:

CreateObject—创建一个 OLE 自动化对象

GetObject—从文件中恢复一个 OLE 自动化对象

CreateObject 函数的语法格式为:

```
CreateObject(class)
```

其中 class 参数是指出用来建立对象和对象类型的应用名字的字符串。使用此函数可创建一个 OLE 自动化对象,并把它赋给一个对象变量。要达到此目的,应首先定义一个对象变量,然后使用 set 语句把由 CreateObject 返回的对象赋给此对象变量。例如:

```
Dim MYOBJect As Object
Set MYOBJect=CreateObject("WordProc.Document")
```

在此代码运行时,若建立对象的应用程序(此例中是 WORDPROC.EXE)还没有运行时,就启动它,并且建立一个指定类型的变量。一旦建立了一个对象,则在 Visual Basic 代码中使用所定义的对象变量。在前面的例子中,可使用对象变量 MyObject 访问它的特性和方法。例如:

```
MyObject.Bold
MyObject.Insert "Hello,World"
MyObject.Print
```

GetObject 函数的语法格式为:

```
GetObject(filename.[,class])
```

参数 filename 包含要恢复对象文件的完整路径和名称。如果 filename 是空字符串,则需要指定 class。使用该函数从文件中访问 OLE 自动化对象,并将对象赋给对象变量。为了这样做,首先要声明对象变量的类型,然后使用 Set 语句将 GetObject 函数返回的对象赋值给对象变量。例如:

```
Dim MyObject As Object
Set MyObject=GetObject("C:\Wordproc\Docs\OLETest.Doc")
```

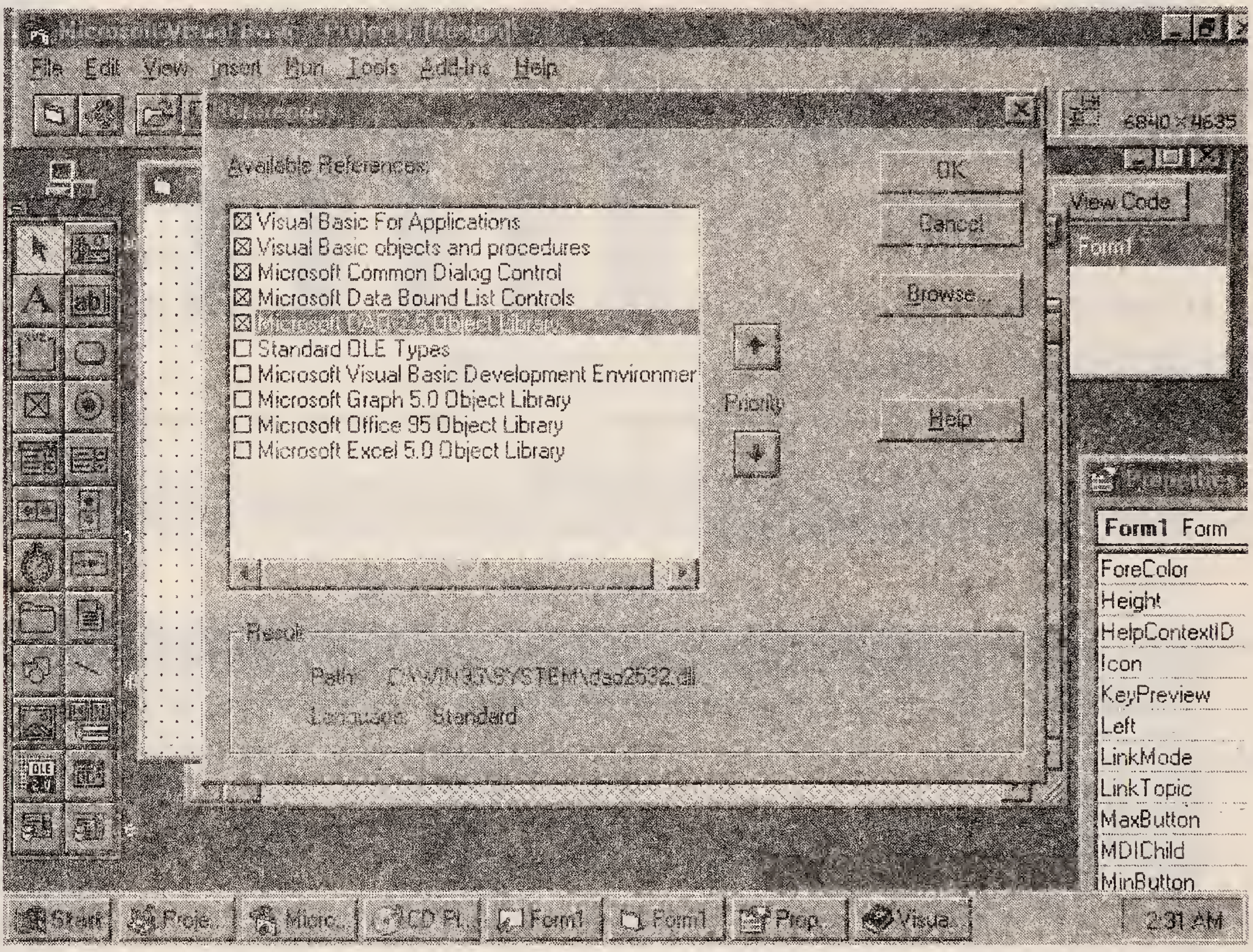
当执行代码时,启动与指定文件名有关的应用程序且激活指定文件中的对象。上面的例子示出了如何激活整个文件,然而,一些应用程序允许激活部分文件。为了做到这一点,在文

件名后加入一个整叹号(!)，这个文件名后跟一个识别要激活的部分文件的字符串。例如，在许多电子表格应用程序中，可使用 R1C1语法指定单元范围的列和行。用户能够用下列代码激活 REVENUE.SPD 的电子表格的单元范围：

```
Set Sheet=("C:\ACCOUNTS\REVENUE.SPD!R1C1:R10R20")
```

一旦激活了对象，便能够在使用所定义对象变量的 Visual Basic 代码中运用它。

最容易得到的对象是那些包括一个对象库的应用程序所提供的对象，为了查阅这些对象可单击 Tools 菜单的 References 选项打开 References 对话框，见图12.40。


File　Edit　View　Insert　Run　Tools　Add-Ins　Help
6840 × 4635
Available References:
OK
Cancel
Browse...
Priority
Help
View Code
Form1
Form1 Form
ForeColor
Height
HelpContextID
Icon
KeyPreview
Left
LinkMode
LinkTopic
MaxButton
MDIChild
MinButton
Start
Proje...
Micro...
CD Pl...
Form1
Form1
Prop...
Visua...
2:31 AM

☒ Visual Basic For Applications
☒ Visual Basic objects and procedures
☒ Microsoft Common Dialog Control
☒ Microsoft Data Bound List Controls
☒ Microsoft DAO 2.5 Object Library
☐ Standard OLE Types
☐ Microsoft Visual Basic Development Environmer
☐ Microsoft Graph 5.0 Object Library
☐ Microsoft Office 95 Object Library
☐ Microsoft Excel 5.0 Object Library

Result
Path:　C:\WIN95\SYSTEM\dao2532.dll
Language:　Standard

图12.40　References 对话框

如果一个对象库已在列表中被列出，选择它名字旁边的复选框。如果没有被列出，用户可以通过单击 Browse 按钮寻找带有.TLB 或.OLB 扩展名的文件。用户也可以寻找带有.EXE 或.DLL 扩展名的文件，因为它们有时也含有对象库。

当用户为一个对象库设置了引用之后，用户能够用对象浏览器查看包含在被引用对象库中的各种对象，同时也能查看同每个对象相联系的属性和方法。单击 View 菜单的 Object Browser 选项或按 F2键打开 Object Browse 对话框，见图12.41。

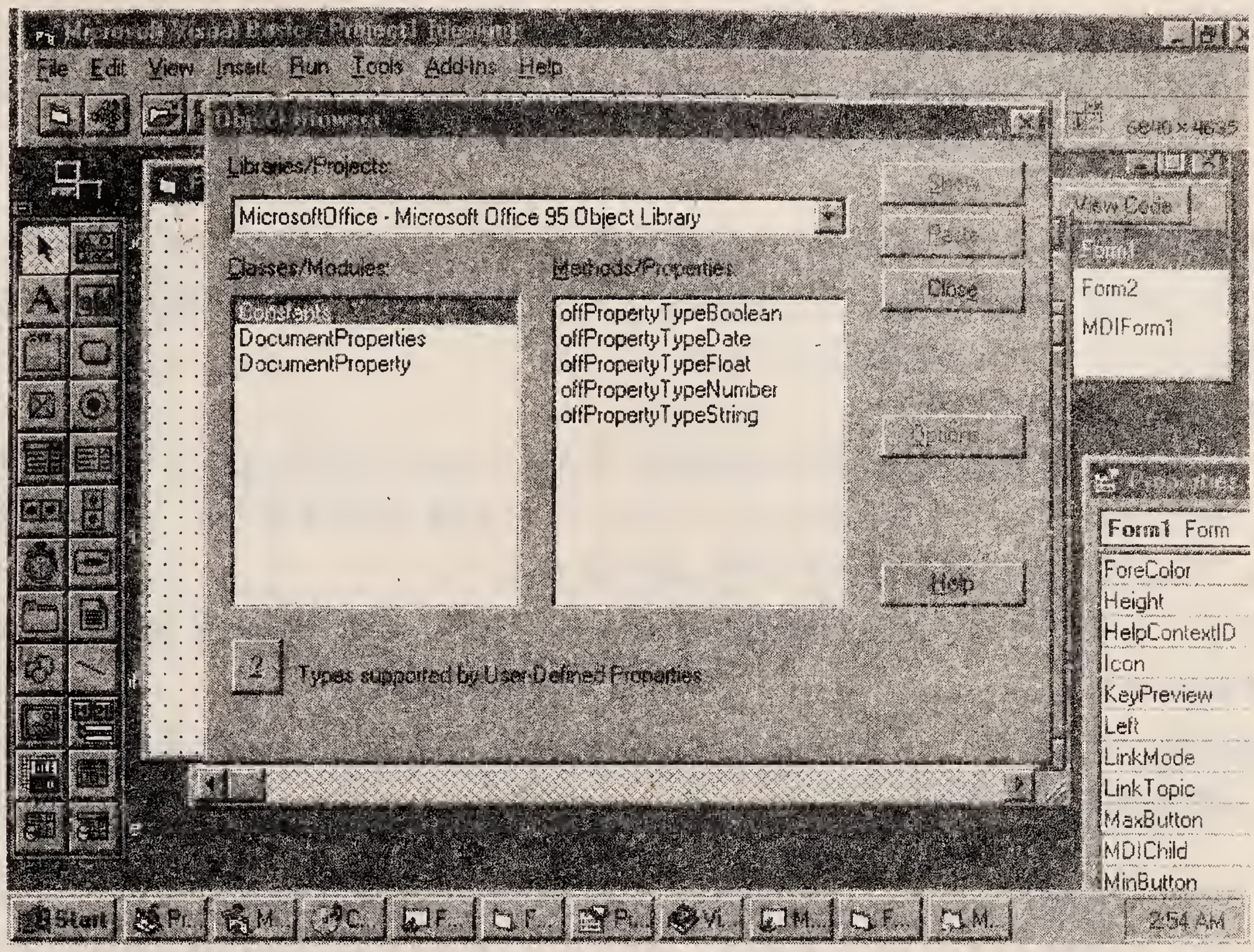
用户可以通过 Libraires/Projects 下拉式列表框选择被引用的对象库，当用户选择了一个对象库以后，Classes/Modules 显示该对象库中可提供的对象。选择了一个对象后，Methods/Properties 列表框显示同该对象相关的属性和方法。如果用户想了解某一属性或方法的帮助信息，可以选取相应的对象，然后单击 Object Browser 对话框左下部的"?"按钮就可获取相应的帮助信息。

为了使用一个 OLE 自动化对象，用户必须完成以下几步：

(1) 为对象变量创建一个存储单元；

(2) 设置变量为一个引用对象或一个已存在的对象；

(3) 用对象的属性和方法编写代码



Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
Object Browser
Libraries/Projects:
Classes/Modules:
Methods/Properties:
Show
Paste
Close
Options...
Help
View Code
Form1
Form2
MDIForm1
Properties
Form1 Form
ForeColor
Height
HelpContextID
Icon
KeyPreview
Left
LinkMode
LinkTopic
MaxButton
MDIChild
MinButton
Start
2:54 AM

Libraries/Projects: MicrosoftOffice - Microsoft Office 95 Object Library

Classes/Modules: Constants, DocumentProperties, DocumentProperty

Methods/Properties: offPropertyTypeBoolean, offPropertyTypeDate, offPropertyTypeFloat, offPropertyTypeNumber, offPropertyTypeString

? Types supported by User-Defined Properties

图12.41 Object Browser 对话框

(4) 通过设置变量为 Nothing 来释放对象

创建存储单元

为了创建一个存储单元需声明一个对象变量，例如：

```
Dim objX As Object
```

设置变量为特定的对象

如果用户需要创建一个新类对象，可以用 CreateObject 函数。例如，下面的例子将会创建一个新的 Microsoft Excel Worksheet 对象：

```
Set objeX=CreateObject("Excel. Sheet")
```

如果对象已存在于一个文件中，用户能够用 GetObject 函数装载它。例如，下面将引用一个已存在的 Microsoft Excel Worksheet 对象：

```
Set ojbX=GetObject("C:\EXCEL\REVENUE. XLS")
```

编写代码

现在用户已经引用一个由另外应用程序提供的对象，用户能够使用这个对象就象使用由 Visual Basic 提供的控件一样。例如，对于 Microsoft Excel wordsheet 的对象 objX，用户能够用 worksheet 的属性和方法增加数据到单元中：

```
objX. Application. visible=True
    For i=1 To 10
        objX. Cells(i,i). Value=i
    Next i
```

在这个例子中，The Microsoft Excel Worksheet 是一个可见对象，对于那些没有可见界面的对象，例如 Data Access 对象，不能够被放置在窗体上，但是通过 Visual Basic 的代码仍然能够操纵和访问它。

释放对象

当用户用对象完成相关的任务后，清除引用对象的任何变量可使对象从内存中释放。为清除一个对象，设置它的变量为 Nothing，例如：

Set ojbX＝Nothing

通过下面的练习用户将创建一个应用程序，该程序用 Excel Spelling Checker 修改在 VisualBasic 文本框中的拼写错误。这个程序完全是在 Visual Basic 中建立起来的，通过 OLE 自动化技术引入了 Excel 的功能。

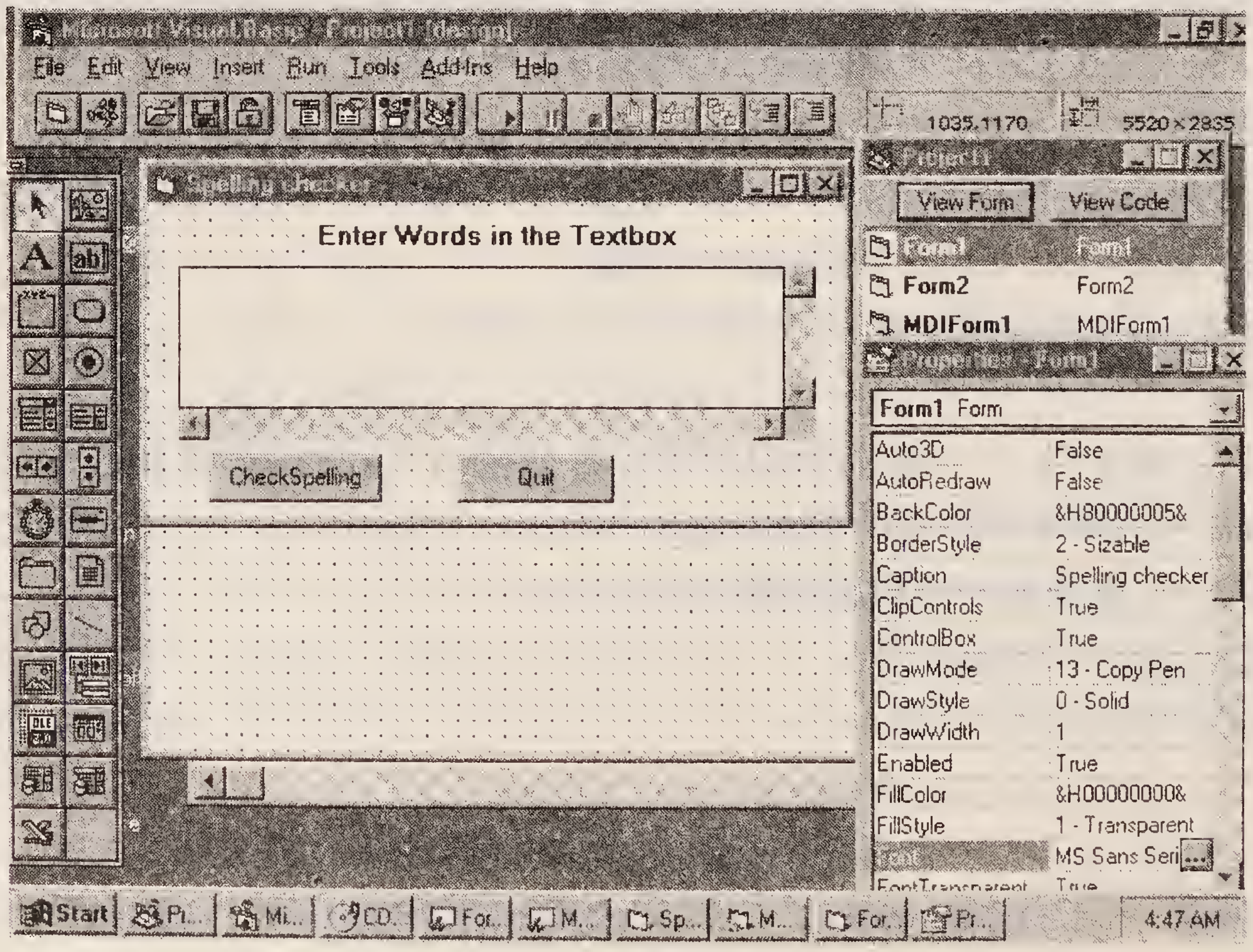
例：创建一个拼写检测器。

(1) 在 Tool 菜单单击 References 选项打开 References 对话框，确定 Microsoft Excel 5.0 或 Microsoft Excel 7.0 for Windows 95 Library 已被选择，然后单击 OK。该对象库允许用户引用由 Excel 对象应用程序释放的对象、属性和方法。

提示　如果用户对将要使用的属性和方法不了解，可以用对象浏览器(Object Browser)获取有关帮助信息。

(2) 把一个文本框，一个标签和两个命令按钮添加到窗体上，设置属性如下表，建立如图12.42所示的初始化界面。

对象	属性	设置
窗体	Caption	Spelling Checker
标签	Caption	Enter Words in the textbox
文本框	Text	空白
	Multiline	True
	ScrollBars	3-Both
命令按钮1	Caption	CheckSpelling
	Name	CmCheck
命令按钮2	Caption	Quit
	Name	CmQuit


File
Edit
View
Insert
Run
Tools
Add-Ins
Help
1035,1170
5520×2835
Enter Words in the Textbox
CheckSpelling
Quit
View Form
View Code
Form1 Form
Start
Pi...
Mi...
CD...
For...
M...
Sp...
M...
For...
Pr...
4:47 AM

Form2	Form2
MDIForm1	MDIForm1

Auto3D	False
AutoRedraw	False
BackColor	&H80000005&
BorderStyle	2 - Sizable
Caption	Spelling checker
ClipControls	True
ControlBox	True
DrawMode	13 - Copy Pen
DrawStyle	0 - Solid
DrawWidth	1
Enabled	True
FillColor	&H00000000&
FillStyle	1 - Transparent
Font	MS Sans Seri

图12.42　初始化界面

(3) 编写代码。双击窗体的空白处打开代码窗口编写以下代码：

```
Sub CmCheck _ Click()
  Dim objX As Object                     'Create as Object variable
  Set objX=CreateObject("Excel. Sheet")
  ObjX. Cells(1,1). Value=Text1. Text    'Copy textbox To cell1,1
  ObjX. Visible=True                     'make dialogbox visible
  ObjX. CheckSpelling                    'use CheckSpelling method
  ObjX. Application. Quit                'Quit Excel
  Set ojbX=Nothing                       'release Object variable
End Sub
  Sub CmQuit _ Click()
    End
End Sub
```

以上语句在事件过程中创建了一个自动化对象，设置了 Excel 对象的两个属性，调用了 Excel 对象的两种方法，最后释放被对象所使用的内存。当引用变量被建立的时候，Excel 自动运行。Excel 单元的 Value 属性被用来拷贝文本框的内容到 Excel Worvksheet 的 A1单元。Visible 方法用来使 SpellingChecker 运行时可见。当 CheckSpelling 方法被执行时，Excel 运行 SpellingChecker 并且检查 Worksheet 的内容。当拼写错误被发现的时候，Excel 的 Spelling 对话框出现。这个对话框允许用户修改错误。当检查过程完成后，正确的单词被拷贝到 Visual-Basic 的文本框中并且 Excel 应用程序关闭。最后一条语句用来释放对象变量的内存。

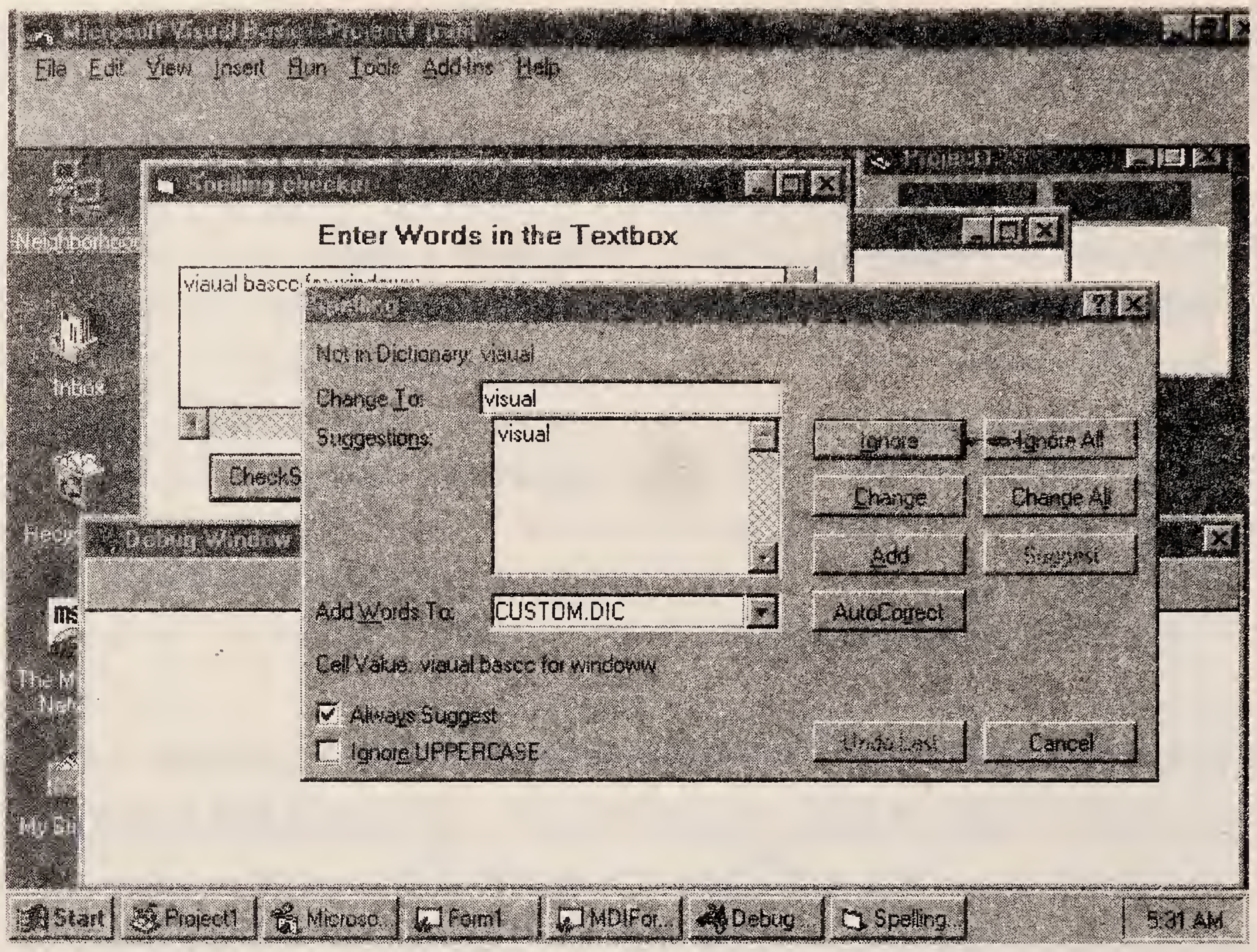
(4) 运行程序。按 F5键或者单击工具条的 Start 按钮，程序开始运行。在文本框中键入下面一句话：

Visual Basic For Windows

单击 CheckSpelling 按钮，Visual Basic 创建一个 OLE 自动化对象并且开始运行 Excel，过一会儿 Spelling 对话框出现，见图12.43。

由图12.43可知第一个单词拼写错了，Spelling 对话框中列出可供选择的正确单词，选择一个正确单词后单击 Change 按钮就可修改此单词。修改完第一个单词之后，Spelling 对话框又列出第二个拼写错误单词的可供选择的修改，用户可以按上述方法修改完所有的拼写错误单词，最后又返回到初始窗体上，并且文本框的内容修改为 Visual Basic for Windows。

(5) 退出程序。单击 Quit 按钮可以结束程序的运行。


File Edit View Insert Run Tools Add-Ins Help
Spelling checker
Spelling
CheckS
Debug Window
Ignore
Ignore All
Change
Change All
Add
Suggest
AutoCorrect
Undo Last
Cancel
Start
Project1
Microso...
Form1
MDIFor...
Debug
Spelling...
5:31 AM

Enter Words in the Textbox

viaual basco for windoww

Not in Dictionary: viaual

Change To: visual

Suggestions: visual

Add Words To: CUSTOM.DIC

Cell Value: viaual basco for windoww

☑ Always Suggest

☐ Ignore UPPERCASE

图12.43 Spelling 对话框

12.5 定制控件概述

Visual Basic 4.0较 Visual Basic 前期版本还有一个特色是它为用户提供了更多的定制控件，这些定制控件使用户工作起来更为方便。所谓定制控件就是指那些带有.VBX 或.OCX 扩展名的文件，或者是那些用定制控件对话框能够被增加到工具箱的可插入对象。象我们前面讲到的公共对话框就是一种定制控件。

12.5.1 定制控件对话框

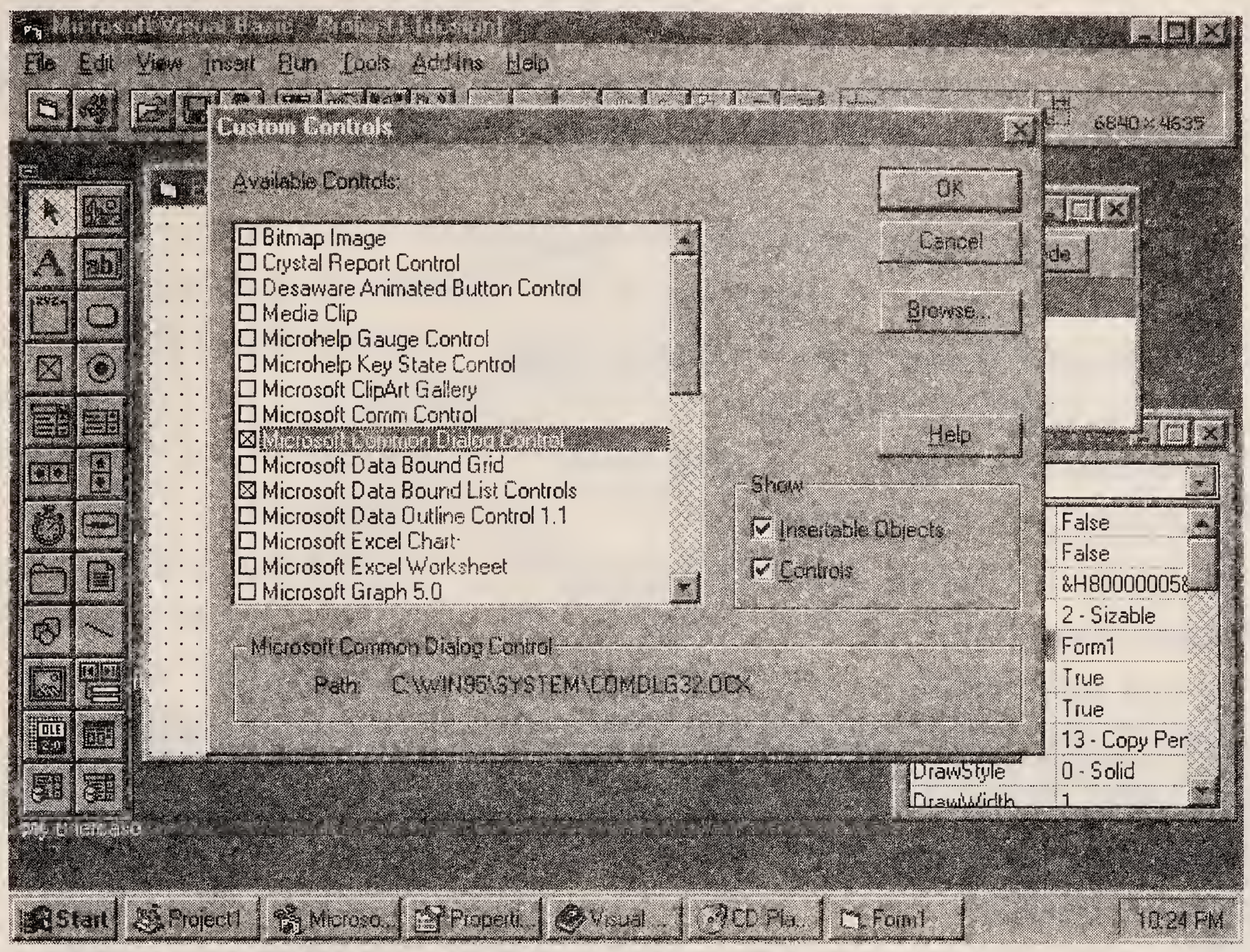
用户能够用定制控件对话框增加一个定制控件或一个可插入对象到工具箱中。所谓可插入对象就是指一个应用程序的对象，例如一个 Microsoft Excel Sheet，它也是一个类型的定制控件。为了打开定制控件对话框，可以单击 Tools 菜单的 Custom Control 选项，见图12.44。

在图12.44中，Available Controls 的列表中显示了用户能够得到的定制控件和可插入对象。为了增加定制控件和可插入对象到工具箱中，选择每项名旁的复选框，然后单击 OK。一旦定制控件被放入工具箱中，就象用户使用标准控件一样也能够把定制控件添加到窗体中。下面列出 Visual Basic 4.0提供的常用定制控件。

Desawave Animated button control

该定制控件是一种具有很大灵活性的按钮控件，它允许用户用图标和位图定义一个用户自己的按钮控件．所定义的控件类型包括激励按钮、多状态按钮和被激励复选框。

Microsoft Gauge Control


Microsoft Visual Basic
File Edit View Insert Run Tools Add-Ins Help
6840 x 4635
Custom Controls
OK
Cancel
Browse...
Help
Start
Project1
Microso...
Properti...
Visual ...
CD Pla...
Form1
10:24 PM

Available Controls:

☐ Bitmap Image
☐ Crystal Report Control
☐ Desaware Animated Button Control
☐ Media Clip
☐ Microhelp Gauge Control
☐ Microhelp Key State Control
☐ Microsoft ClipArt Gallery
☐ Microsoft Comm Control
☒ Microsoft Common Dialog Control
☐ Microsoft Data Bound Grid
☒ Microsoft Data Bound List Controls
☐ Microsoft Data Outline Control 1.1
☐ Microsoft Excel Chart
☐ Microsoft Excel Worksheet
☐ Microsoft Graph 5.0

Show
☑ Insertable Objects
☑ Controls

Microsoft Common Dialog Control
Path: C:\WIN95\SYSTEM\COMDLG32.OCX

False
False
&H80000005&
2 - Sizable
Form1
True
True
13 - Copy Pen
DrawStyle 0 - Solid
DrawWidth 1

图12.44 定制控件对话框

该定制控件可以创建一个用户自定义的标准尺寸。

Microsoft Data Bound List Controls(DBList)

该定制控件能够用一个数据控件中的字段自动填充一个数据约束列表框，能够把被选择的字段传递到第二个数据控件。

Microsoft Data Bound Combo Controls(DBCombo)

该定制控件能够用一个数据控件中的字段自动填充一个数据约束组合框，能够把被选择的字段传递到第二个数据控件。用户还可以在组合框的文本字段编辑被选择的字段。

Microsoft Masked Edit Control

该定制控件象格式化数据输出一样能够提供限制数据输入的功能。它能够提供关于被键入或被显示数据类型的直观的提示。

Microsoft Multimedia Control

该定制控件能够管理多媒体文件的记录和信息的反馈。

Microsoft Outline Control

该定制控件是一种特殊类型的列表控件，它允许用户按水平列表方式显示列项。这个控件对于在文件系统中显示目录和文件特别有用。

Microsoft Picture Clip Control

该定制控件允许使用一个源位图中的一部分，然后在窗体或图片框中显示该部分的图像。该控件在运行时是不可见的。

Outrider SpinButton Control

该定制控件是一个旋转控件，用户能够用另一个控件增加或减少它的数值。用户也能够用它在一定的数值范围内向前或向后滚动。

Pinnacle-Bps Graph Control

该定制控件允许用户在窗体上设计相互影响的图表。在运行时，用户能够发送新的数据到图表中，重画它们、打印它们、把它们拷贝到剪贴板上或者改变它们的风格和形状。

Sheridon 3D Controls

把该定制控件加到工具箱中可以出现以下定制控件：

(1) 3D Check box。显示一个三维复选框

(2) 3D Command Button。显示一个三维命令按钮

(3) 3D Frame。显示一个三维框架

(4) 3D Option button。显示一个三维单选按钮

(5) 3D Panel。显示一个三维面板

(6) 3D group push button。显示一组三维按钮

Microsoft Grid Control

该控件用来显示一系列的行和列，并且操纵在它每个单元中的数据。

Microsoft Data Bound Grid

该控件用来显示一系列的行和列并且操纵在它每个单元中的数据。它拥有标准网格控件所没有的增强的数据访问能力。

Microsoft Comm Control

通过一个串行端口传递和接收数据，该控件为用户的应用程序提供了串行通信的功能。

Microsoft Key Control

用户能够用这个定制控件控制 CapsLock、NumLock 和 SCROLL LOCK 的键盘状态的显示或者修改它们的键盘状态。

12.5.2 DBList 和 DBCombo 定制控件的应用

DBList 和 DBCombo 控件分别自动填充一个数据约束列表框和一个数据约束组合框。它们能够用一个数据控件的字段来自动添加一个数据到列表框中，或者用来传递一个被选择的字段到第二个数据控件中。

数据约束列表框(DBList)和数据约束组合框(DBCombo)不同于标准的列表框和组合控件，因为它们能代替用 Additem 方法自动用来自记录集对象的数据填充列表框和组合框。另外，DBList 和 DBCombo 定制控件支持自动寻找模式。下表列出了一些属性，用户能够用这些属性填充和管理列表，约束被选择的属性到一个数据控件。

属性	描述
DataSource	确定通过哪个数据控件把当前控件与数据库联系起来
DataField	确定由 DataSource 属性所确定的记录集中被更新的字段名
RowSource	指定为控件列表项提供数据的数据控件名
ListField	由 DataSource 属性所确定的记录集中的字段名，它被用来填充列表。通常是一个描述字段被用，而不是一个代码字段被用
BoundColumn	确定由 RowSource 属性所指定的记录集中的一个字段名，一旦做出一个选择，该字段将返回给 DataField 属性
BoundText	BoundColumn 属性所指定字段的文本值，在做出一个选择之后，该值返回给 DataSource 属性用来更新 DataField 属性的字段

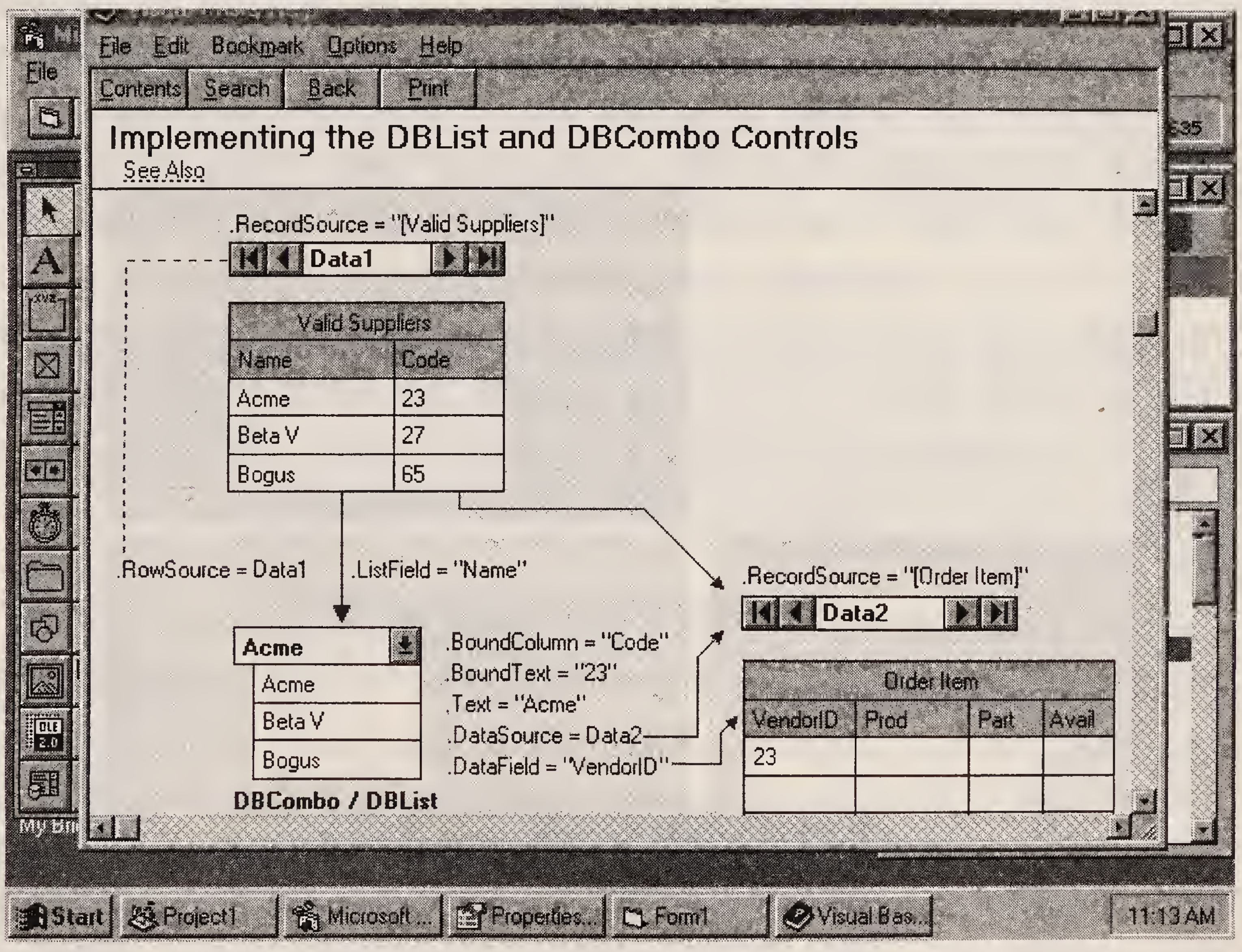
DBList 和 DBCombo 控件使用方法基本相同。DBList 和 DBCombo 控件有两种模式，它们能够被单独使用或同时使用。

AutoFill 模式：用所选择的字段自动填充列表，该字段来自由 RecordSource 属性所确定的数据控件所管理的所有记录。

AutoUpdate 模式：约束在控件中所选择的记录到特定的字段中，该字段存在于由 DataSource 属性确定的数据控件所管理的记录集中。

用户能够用这两种模式建立一个基于数据库上的项目列表。当列表中的一项被选择时，它能够被由 DataSource 属性所确定的数据控件得到，该选择项也出现在控件的文本框部分，在文本框中该选择项能够被用户编辑。如果用户在文本框部分键入一个值，控件的列表部分试图准备一个位置匹配该项。如果一个匹配项被发现，BoundText 属性被设置(根据由 BoundColumn 属性所确定的字段)。如果没有匹配项被发现，BoundText 属性被设置为 Null。

像图12.45的 DBCombo 控件一样，用户能够从项目列表中选择一项，并把选择的列项或由 RowSource 属性所确定的记录集中的字段传递到另一个数据控件中。


File
Edit
Bookmark
Options
Help
Contents
Search
Back
Print
Implementing the DBList and DBCombo Controls
See Also
Data1
Data2
DBCombo / DBList
Start
Project1
Microsoft ...
Properties...
Form1
Visual Bas...
11:13 AM

.RecordSource = "[Valid Suppliers]"

Valid Suppliers	
Name	Code
Acme	23
Beta V	27
Bogus	65

.RowSource = Data1

.ListField = "Name"

Acme
Acme
Beta V
Bogus

.BoundColumn = "Code"
.BoundText = "23"
.Text = "Acme"
.DataSource = Data2
.DataField = "VendorID"

.RecordSource = "[Order Item]"

Order Item			
VendorID	Prod	Part	Avail
23			

图12.45 DBCombo 定制控件使用示图

一旦数据控件移动另一个记录或者被更新，来自 BoundColumn 属性的设置值被传递给由 DataSource 属性所确定的控件中。如果没有相匹配的项，则一个 Null 值被传送给数据控件。

例如，在图12.45中，用户能够用 DBCombo 控件来纠正和显示当前提供的列表。为了更新由 OrderItem 表的记录集中的 VendorID 字段，用户能够在 DBCombo 定制控件的列表中选择一个列项或者在 DBCombo 控件的文本框部分键入新的 VendorID。为了完成这个例子，用户必须完成以下步骤：

（1）添加两个数据控件和一个 DBCombo 控件到窗体上。

（2）设置第二个控件的相关属性以便用 OrderItem 表创建一个记录集。

（3）为了能用 ValidSuppliers 表创建一个瞬态记录集，设置第一个数据控件的属性。为 DBCombo 控件能够管理记录集，它的 Visuable 属性应设置为 False。

（4）设置 DBCombo 的 DataSource 属性为 Data2。

（5）设置 DBCombo 的 DataField 属性为 VendorID 字段。这将约束 DBCombo 控件到 OrderItem 所创建的记录集，同时指定一旦用户做出选择，VendorID 将被更新。

（6）设置 RowSource 属性为 Data1，该数据控件用来填充来自 ValidSuppliers 记录集的列表。

（7）设置 List Field 属性为 Name 字段。这将指定哪个字段用来填写 DBCombo 的列表部分。

（8）设置 Bound Column 属性为 Code 字段。这将约束 DBCombo 控件中被选择的项到第一个数据控件的记录集的字段中，这种情况就象由 DataField 属性指定的一样。在此例中，DataField 属性指向 VendorID 字段，当第一个数据控件被更新，VendorID 字段也被更新。

当用户运行这个程序时，数据约束组合框用由 RowSource 和 ListField 属性所确定的记录集中的当前记录填充列表。在上例中是 Data1的 Name 字段中的记录。用第二个数据控件选择 OrderItem 记录中的一个。接下来，用户从数据约束组合框中选择 ValidSuppliers 表中的一项或者在控件的文本框部分改变被选择的项。然后把第二个数据控件重新定位于不同的记录，最后又返回到原始记录。用户会发现 VendorID 字段被从数据约束组合框中选择的 VendorID 值所更新。如果用户在文本框中提供的值在列表中不存在，则这时会使用一个空字符串来更新。有关定制控件的更多知识，用户可以参考《Visual Basic 4.0用户手册》。

12.5.3 可插入对象的基本使用方法

Visual Basic 4.0除了提供了一些非常有用的定制控件以外，还提供了一些可插入对象，它们可以当定制控件使用。当用户在定制控件对话框选取一个可插入对象，然后单击 OK 按钮时，就可把该插入对象加入到控件工具箱中。例如，选取 Microsoft Word Document，然后单击 OK 按钮，稍候该对象被加入到工具箱中，这时用户可以像使用 Visual Basic 提供的标准控件一样使用它。用户双击工具箱中该对象的图标就可以把它添加到窗体中，见图12.46。

在图12.46中的文本编辑区用户键入文本，并可以用窗体上顶部的菜单命令进行文档的格式化。例如，用户可以单击 View 菜单中的 Toolbars 选项增加一个标准工具条，用这个标准工具条用户可以方便地进行文档的简单格式化工作，见图12.47。

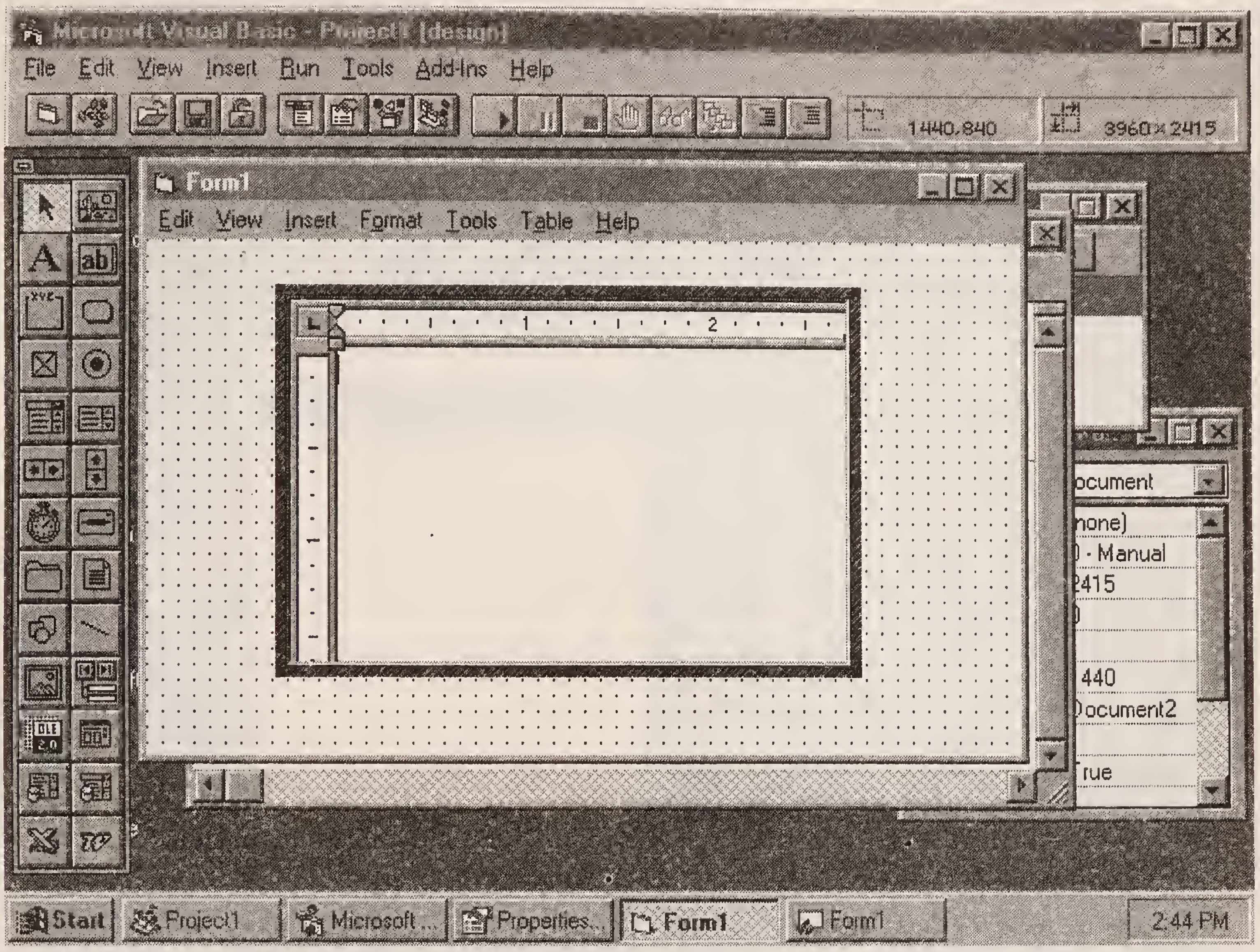
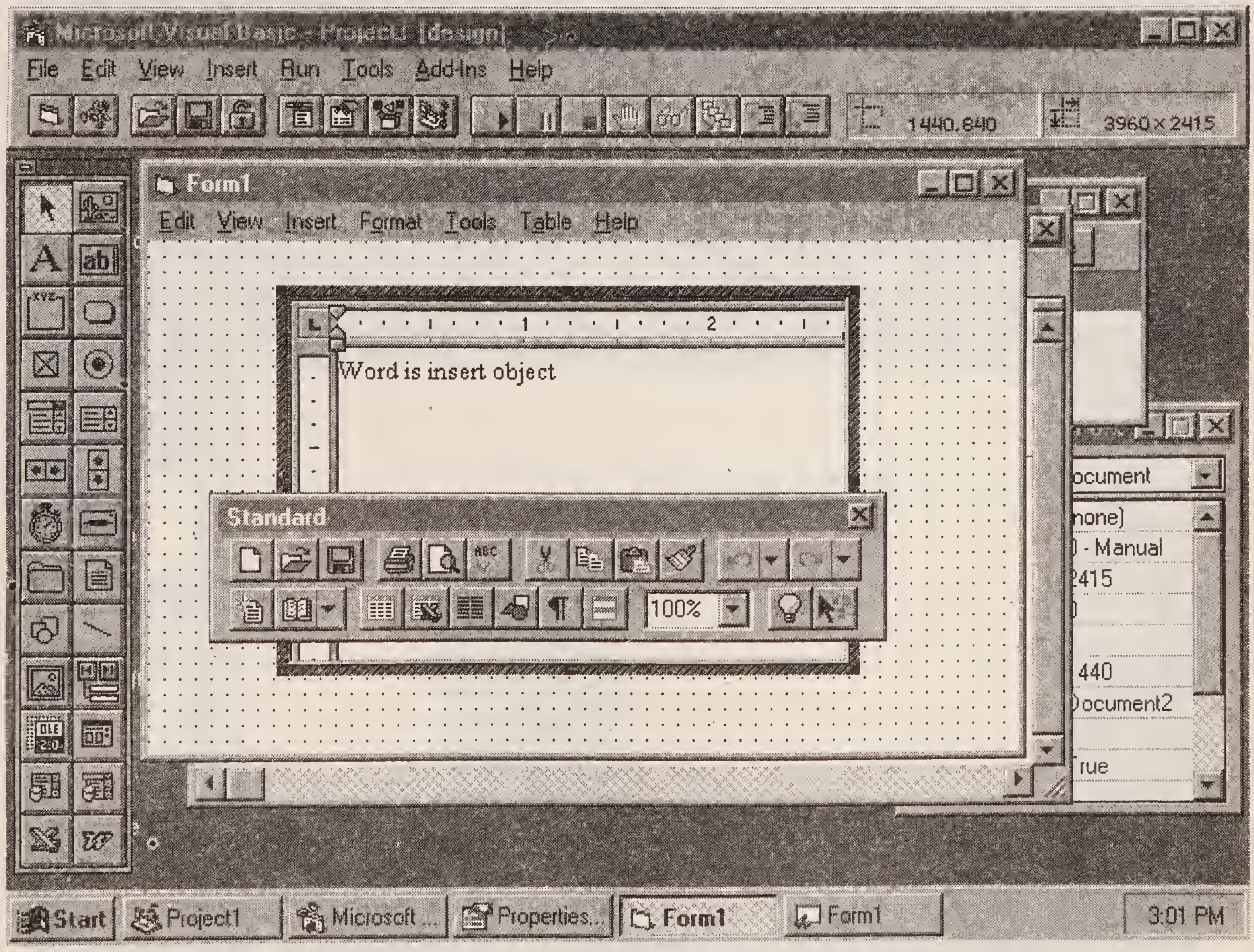

Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
1440,840
3960×2415
Form1
Edit View Insert Format Tools Table Help
ocument
none)
) - Manual
2415
440
Document2
True
Start
Project1
Microsoft ...
Properties...
Form1
Form1
2:44 PM

图12.46　添加到窗体中的 Word 文档对象


Word is insert object

Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
1440,840
3960×2415
Form1
Edit View Insert Format Tools Table Help
Standard
100%
ocument
none)
) - Manual
2415
440
Document2
True
Start
Project1
Microsoft ...
Properties...
Form1
Form1
3:01 PM

图12.47　添加的标准工具条

当用户完成文档的有关操作后，单击窗体上的空白处，Word 文档编辑环境消失，见图12.48。

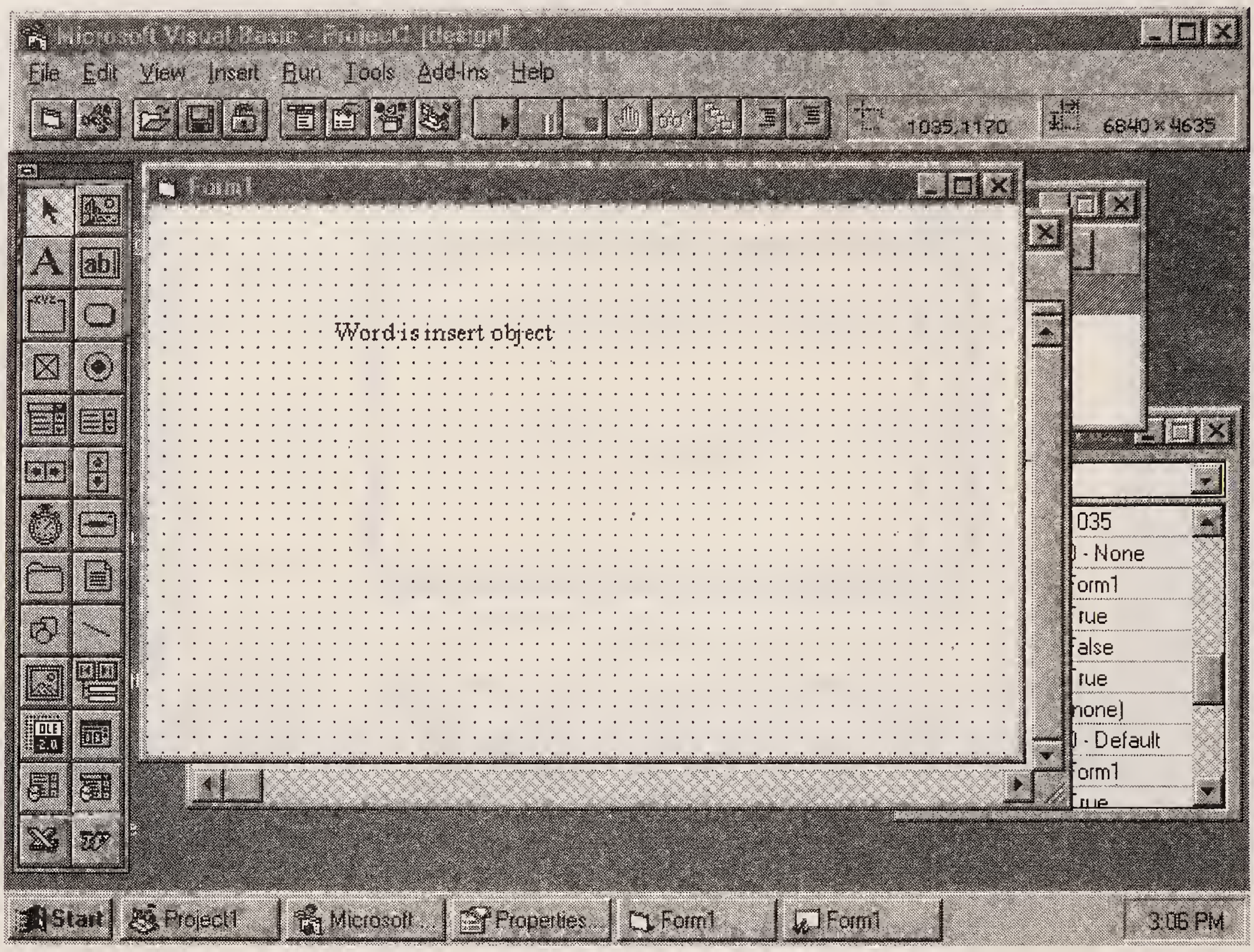

Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
1035,1170
6840 x 4635
Form1
Word is insert object
035
0 - None
Form1
True
False
True
(none)
0 - Default
Form1
True
Start
Project1
Microsoft ...
Properties...
Form1
Form1
3:06 PM

图12.48 **消失编辑环境的 Word 文档**

如果用户想修改文档，可以用鼠标左键激活该文档，然后在文档编辑区用鼠标右键单击，这时屏幕上弹出一菜单，见图12.49。

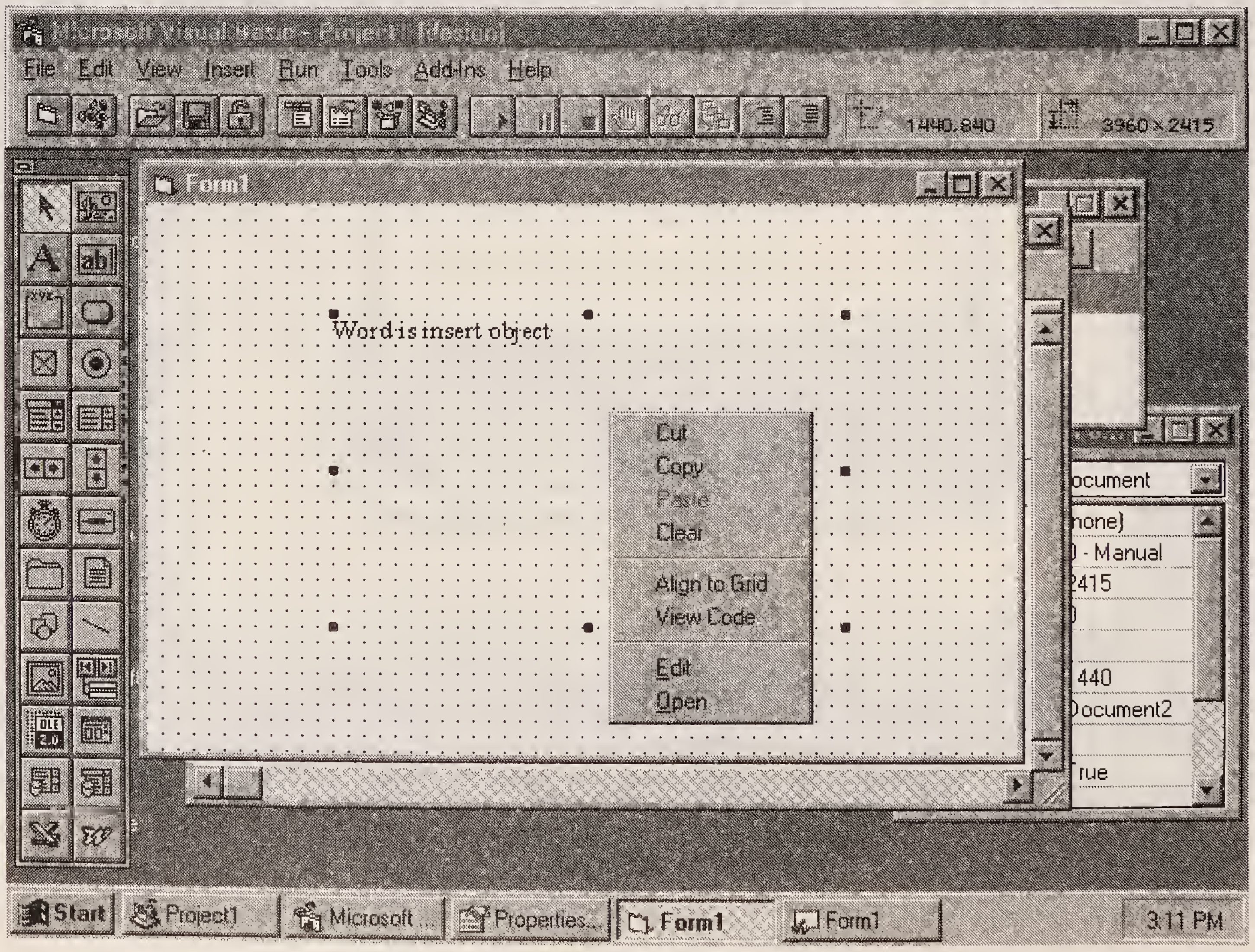
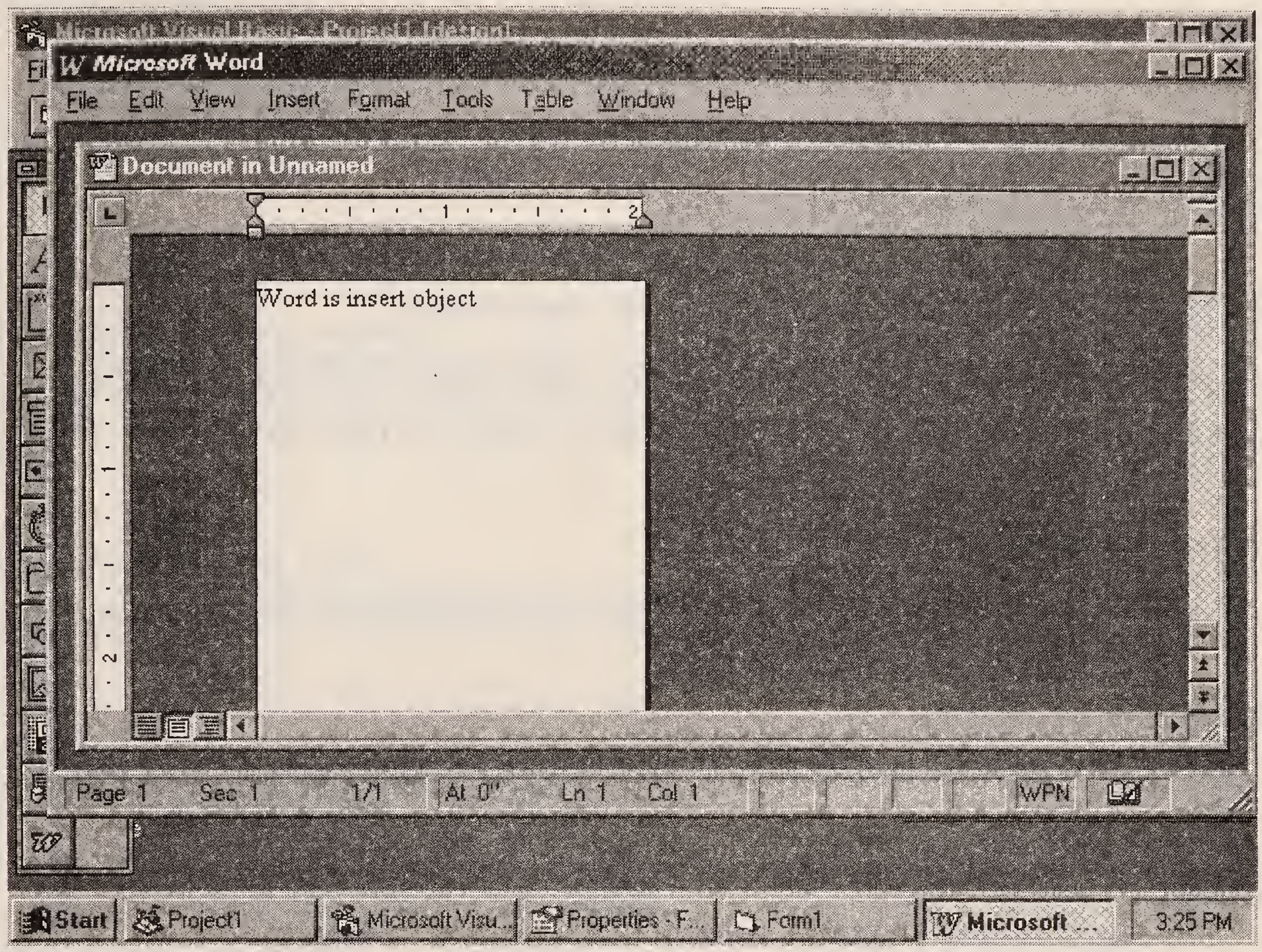

Microsoft Visual Basic - Project1 [design]
File Edit View Insert Run Tools Add-Ins Help
1440,840
3960 x 2415
Form1
Word is insert object
Cut
Copy
Paste
Clear
Align to Grid
View Code
Edit
Open
Document
(none)
0 - Manual
2415
0
1440
Document2
True
Start
Project1
Microsoft ...
Properties...
Form1
Form1
3:11 PM

图12.49 按鼠标右键弹出的菜单

在图12.49的菜单中单击 Edit 选项就可以返回到 Word 文档的编辑环境中。如果用户单击 Open 选项，这时就可启动 Word 应用程序，见图12.50，用户这时也可以进行文档的有关操作，完成操作后可用 File 菜单上的 Exit 选项退出 Word 应用程序。


Microsoft Visual Basic - Project1 [design]
Microsoft Word
File
Edit
View
Insert
Format
Tools
Table
Window
Help
Document in Unnamed
Page 1
Sec 1
1/1
At 0"
Ln 1
Col 1
WPN
Start
Project1
Microsoft Visu...
Properties - F...
Form1
Microsoft ...
3:25 PM

Word is insert object

图12.50　通过启动的 Word 应用程序编辑文档

Visual Basic 4.0除了提供了 Microsoft Word Document 可插入对象外，还提供了以下可插入对象：Microsoft Excel Chart，Microsoft Excel Worksheet 等，它们的使用方法同 Microsoft Word Document 基本相同。

附录　Visual Basic 4.0术语

386增强模式(386 enhanced mode)

Windows 3.x 的三种运行模式之一。这种模式运行在 x86(80386、80486、Pentium P5、Pentium P6)微处理器上,可以访问扩展内存并且支持非 Windows(DOS)应用程序,标准模式和386增强模式运行在保护模式之下。

数学协处理器的8087家族(8087 farmily of math coprocessors)

执行高速浮点运算和把二进制代码转化为十进制运行的处理器,这指的是8087、80287和80387协处理器(大多数80486和 Pentium P5微处理器都有内置的数学协处理器)。

活动窗口(active window)

下一次输出所定向的屏幕上的窗口。活动窗口具有高亮度的标题和边框。

聚集函数(aggregate function)

执行数字运算的函数。聚集函数包括 SUM()、AVG()、MIN()、MAX()和 COUNT()。

警告(alert)

提示一条报警或错误信息的窗口。一些人认为响铃也是一种警告。

别名(alias)

当某个变量处于使用状态时,分配给变量的一个名称。

AND

二进制逻辑运算符。当两个操作数均为真值时,此运算的结果为真;当两个操作数其中之一为假或者两个操作数均为假时,此运算的结果为假。

ANSI(美国国家标准研究所)

民间的研制国家技术标准的组织。

应用编程接口(API,Application Programming Interface 的缩写)

为在应用程序和操作系统之间的操作生成接口的一组例程和函数,这些操作有输入、输出、文件管理和图形输出等。API 的例子有 Microsoft Windows API 和 Visual Basic API。

应用程序(application)

由 Visual Basic 的程序、窗体、菜单和文件组成的、可编译成一个单一文件的文件。

参数(argument)

传递给函数或过程的值。

算术转换(arithmetic conversion)

将数据从数字值转换成另一种类型的一个操作。而在将数字转换成字符型字符串时使用的是Str()函数。

算术表达式(arithmetic expression)

由数字、数字变量、函数和数字操作符组成的表达式。例如X * 3+SQRT(Y)就是一个算术表达式。

算术操作符(arithmetic operator)

代表操作符对数字算子执行某项算术运算的一个符号。

数组(array)

通过某个排列组织起的数据元素的集合。数组中的每个元素都可以用一个数字下标来引用。在Basic中,数组可以有一个或多个下标。

ASCII(美国标准信息交换)

被普遍接受的一组255个字符代码,这些代码代表数字、字符和特殊符号。

汇编器(assembler)

把用符号汇编语言写的程序转换成计算机可以执行的目标代码。

汇编语言(assembly language)

一种低级计算机语言,其中的符号代码与计算机实际执行的机器语言指令具有一对一的对应关系。

后端数据(back-end data)

存在于Visual Basic之外的但由OLE支持的表中的所有数据。后端数据源的例子包括Paradox、Microsoft Access和SQL Server。

基类(base class)

在Visual Basic中,基类是在Visual Basic内部定义的类,用户可以将其作为检索用户定义的类的基准。例如,用户可以从Visual Basic窗体和控件基类中检索出自己的图形对象。

二进制(binary)

一种双状态、开/关式的逻辑。二进制数字计数系统使用1和0来代表数值。

二进制文件(binary file)

一种非文本文件，包含文本字符之外的值。例如 EXE、BMP 和 OBJ 文件。在 Visual Basic 中，一个二进制文件(BIN)包含可执行代码，可以用 LOAD 命令装入内存以及可以用 CALL 命令执行。

二进制格式(binary format)

不经任何转换，存储在内存中的数据可以直接存入磁盘文件。数字型数据用二进制格式的二进制值来代表，而不是一系列的 ASCII 数字。

二进制模式(binary mode)

不执行任何转换来直接访问文件的一种方式。

二进制资源文件(binary resource file)

来自资源编译器(Resource Compiler)的带有.RES 扩展名的输出文件。二进制资源文件包含编译过的 Windows 位图、图标、对话或光标资源数据。

位(bit)

一个二进制位或一个字节的1/8。一个位的值为0或1。位是计算机可处理的最小单位。

位图(bitmap)

存储在带有.BMP 文件扩展名的文件中的代表图像中的象素的位组成的数组。

位图化字体(bitmapped font)

一种字体，其中是用位的数组而不是用一组方程式来代表每个字符。Windows 的 TrueType 字体是由一组方程定义的。

按位操作符(bitwise operator)

该操作符是作用于值的单个位，而不是整个值。在 Visual Basic 中，位方式的操作符是 AND、OR、XOR 和 NOT。

BOF(文件头)

当文件指向文件首时，称为位于文件头。

布尔数(boolean)

代表真/假(True/False)或者是/否(yes/no)的数据。逻辑型数据就是布尔数。

控件(bound control)

在窗体上的一个控件，它从某个窗体或查询中的特定属性和变量中分离出自己的值。例

如，包含当前控件的City属性的文本框就是一个控件。

浏览(browse)

观察列表表格中的数据。当用户观察在Browse表中的信息时，可以浏览表。

缓冲区(buffer)

保留用于当前正在处理的数据的计算机内存的一部分。例如，在输入/输出操作中，缓冲区用来临时存放数据。

字节(byte)

用来访问内存中的地址的8-位式的单位。字节是内存和磁盘空间大小的度量单位。一个字节包含一个ASCII字符。

标题(caption)

窗体的标题或者在控件上的名称。例如，Cancel是Cancel按钮控件上的Caption属性。

区分大小写(case sensitivity)

用来指示是否将大写字母看作是与其对应的小写字母具有相同的值的一种状态。C和UNIX是区分大小写的，Visual Basic、FoxPro、Xbase和MS-DOS是不区分大小写的。

字符(character)

字母符号(例如，A、1、B、2、z)。

字符表达式(character expression)

由字符串、字符型操作符、函数和字符型变量组成的表达式。

字符串(character string)

作为ASCII字符(字母、数字和其他符号)处理的连续字节。在Basic中，字符串经常括在单引号、双引号或方括号中(例如，“string”，’string’，[string])。

复选框(check box)

菜单控件元素。它是后面跟有文本标题或图形的一个小方框。通过单击复选框，可以选中或关闭复选框。当其为选中状态时，框中会显示一个X；当其为关闭状态时，框中为空。

类(class)

定义对象的属性以及定义对象的外观和动作的模板。

语句(clause)

在Visual Basic中，跟在命令谓词后面的语言成分。语句可以是一个关键字，或者是后面

跟有一些其他语言成分的一个关键字，例如表达式或变量。

单击(click)

当鼠标指针指向屏幕上的某个对象时，按下并迅速释放鼠标按钮之一，最常用的是左鼠标按钮的动作，称为单击。

客户-服务器应用程序(client-server application)

从远程服务器访问数据的本地的(客户)用户接口应用，在客户和服务器之间协调工作。

剪贴板(clipboard)

文本的临时存储缓冲区。剪贴板用来剪切、粘贴和拷贝文本。

代码页(code page)

操作系统用来定义字符集的一个表。字符以数字值的形式存储在内存中。当 Windows 显示一个数字值时，它使用一个特定的代码页来将数字值映射到专门的字符集中。不同的国家使用不同的字符集。

代码符号(code symbol)

某个过程或函数的地址。

命令(command)

用户向计算机发布的语句，使计算机执行某类行为。在 Basic 语言中，命令常常由一个谓词以及一些附加的语句组成。用户可以在程序中加入命令，或者在控件的方法中键入命令。在 Windows 中，一个菜单选项经常称为一个命令。

命令关键字(command keyword)

指定某些活动的命令的成分。某些关键字是独立的，其他一些关键字包含有语句。

注释(comments)

非执行程序语句。注释常用来提供一些有关程序是做什么的信息。在 Basic 中，跟在’分界符后面的文本为注释。在 C 语言中，注释由/ * 和 * /来分界。

编译(compile)

将编程语言源代码翻译成可执行代码的过程。

编译器(compiler)

将编程语言源代码翻译成可执行代码的计算机程序。Basic 中含有一个内置的编译器，用来将用 Basic 语言写的程序文件编译成 Basic 译码器可以执行的伪代码。

编译器指令(compiler directive)

用于程序中的、在程序编译前要处理的特殊命令。编译器指令有时称为预处理指令。编译器指令的第一个字符为＃。参见指令(directive)。

条件表达式(conditional expression)

求值结果为真(.T.)或假(.F.)的语言表达式。

常数(constant)

在执行过程中不改变的一个值。常数的例子有345.43,“This is a constant”和{12/12/94}。

常数表达式(constant expression)

求值结果为常数的表达式。

包容器类(container classes)

组合起其他类的 Visual Basic 基类。

控件(control)

带有简单的输入输出功能的图形对象。例如按钮、编辑框、标签、直线、栅格和文本框。

控件数组(control array)

共享相同的名称、属性和事件过程的一组控件。每个控件都有唯一的下标。当控件之一对事件作出反应时,它就调用向其中传递了该控件的唯一下标值的事件过程。

控制符(control character)

用来编辑控件或快捷键的非打印字符。输入控制符时,可以按住 Ctrl 键加上一个字符键。控制键序列的符号是 Ctrl+<字符>。例如,Ctrl+C。

协处理器(coprocessor)

参见数学协处理器的8087家族。

CPU

中央处理器单元。

标准(criteria)

用在查询或过滤器中指定一组文件的一组限制条件。

.CUR

光标资源对象的文件扩展名。

光标(cursor)

指示键盘输入位置或鼠标动作的一条闪动的细线或其他代表符号。

定制控件(custom control)

基于一个或多个 Visual Basic 基类的用户创建的控件。

数据管理器(Data Manager)

Visual Basic 4.0的一个工具,用来对数据库进行管理。

数据类型(data type)

用来定义在 Visual Basic 中数据按何种方式进行存储和维护的术语。Visual Basic 中的9种数据类型为 Any 数据类型、货币型、双精度型、整型、长整型、单精度型、字符串型、用户定义类型和变体型。

数据库(database)

相关的数据和对象的集合。

数据库管理系统(DBMS,DataBase Management System 的缩写)

用来组织、分析、存储、检索和编辑信息的软件。

动态数据交换(DDE,Dynamic Data Exchange 的缩写)

用来在 Windows 应用程序之间交换数据的已建立好的协议。如果在运行的两个 Windows 应用程序可使用剪切和粘贴操作,则可以在它们之间交换数据。

调试器(debugger)

用来检测错误的程序开发工具,例如 Visual Basic Debugger。

声明(declaration)

将名称与其他属性——例如类型或值——结合起来的一种语言结构。

降序(decrement)

按逆序记数(例如,7、6、5、4)。

缺省(default)

未经指定但已由程序假定的值和行为。例如,如果用户不指定值,计算机就使用缺省值。

缺省控件(default control)

在一组控件中(单选按钮、按钮等等),缺省控件是预先选择的那个控件。如果不选择任

何控件就按 Enter 键，选取的就是缺省控件。

定义(definition)

为变量指定并且初始化内存需求或为函数指定属性的语言结构。

分界符(delimiter)

标志某个单元的数据的开头和结尾的字符。数据单元是一个相关的字符序列。例如数字或关键字。单引号、双引号或方括号都是字符串常量的分界符(例如“Aardvark”)。

设计阶段(design time)

用户创建应用程序的一段时间。在设计阶段中，用户可以建立组成部分以及设置属性。某些属性可以在设计阶段中设置，另一些属性可以在运行阶段中设置。

桌面(desktop)

Visual Basic 主窗口，所有的 Visual Basic 系统、用户定义的窗口和其他对象都显示在这个工作空间上 。

对话框(dialog box)

显示程序信息或提示用户输入的临时窗口。

维数(dimension)

数组下标的个数。

指令(directive)

让预处理器在编译前对源代码执行某些操作的指示。

禁止(disable)

使某个对象不能使用。例如 Windows 程序中被禁止的菜单选项，这些菜单选项常以较暗的颜色显示，并且不能选取。

DLL(动态链接库，Dynamic Link Library 的缩写)

使应用程序共享资源和功能的特定库。为 Visual Basic 开发的 DLL 文件称为 Visual Basic 链接库文件(VBX)。

锚定的(docked)

对象可以缩小放在 Visual Basic 主窗口的顶部两侧或底部。工具栏是可锚定对象的例子。

双击(double-click)

当鼠标指向屏幕上的某个对象时，迅速连续两次按下并马上释放某个鼠标按钮的动作。经常使用的是左鼠标按钮。

双精度(double precision)

占用8字节内存的浮点数字值。双精度值精确到15或16位有效数字。

DPMI(MS-DOS 保护模式界面)

使程序可以利用扩展或扩充内存的一个服务程序。DPMI 服务程序的例子包括 DOS 会话的 MSDPMI 处理程序，此服务程序可以允许程序使用80386微处理器的32位体系结构。Microsoft C/C++ 7的编译器就是一个32位的程序，除非正在运行 MSDPMI，否则它不能运行在 DOS 下。该程序也可以运行在 Windows 下的 DOS 会话中。

拖动(drag)

按住鼠标按钮移动鼠标，当到达目标位置后释放鼠标按钮，就完成了拖动操作。

拖放(drag and drop)

拖动某个对象到另一个对象上执行某个处理的操作。

动态数据交换(DDE,Dynamic Data Exchange)

参见 DDE。

动态链接库(dynamic link library)

参见 DLL。

动态链接(dynamic linking)

在运行阶段为某个应用程序执行连接操作的过程。

EGA(增强图形适配器)

IBM PC 及兼容机的彩色视频显示设备，可显示64种颜色，最多同时显示16种颜色。

元素(element)

数组的一个独立成员。

嵌入(embed)

OLE 中的一个术语，用来指示来自一个应用程序中的对象的拷贝被插入到另一个应用程序中，二者在此后不再有任何联系。对拷贝的修改不会影响原来的对象。

仿真器(emulator)

模拟数字处理器操作的软件包。

封装(encapsulation)

隐藏起某个模块的内部活动并支持提取的一种结构。

环境变量(environment variable)

存储在 MS-DOS 环境表中的变量，此表包含有与软件环境有关的信息，环境变量最常用来存储信息以帮助寻找文件。

EOF(文件尾)

指示文件尾的代码。

转义序列(escape sequence)

前面加有一个代码扩展字符(也称为转义字符)的几个字符的组合。代码扩展字符表示后续的字符要另外解释。转义序列通常用来控制打印或显示输出。Esc 键有时用作转义字符。

事件(event)

对象能够识别的一项活动，例如鼠标单击。当事件发生时，执行事件代码。

事件代码(event code)

当事件发生时执行的代码。

事件驱动(event driven)

发生某些事件时触发某个活动的条件，例如鼠标单击或按键。

.EXE

可执行应用程序文件的扩展名。

可执行文件(execuable file)

包含机语言指令，可以从操作系统启动运行的程序。

执行(execute)

运行软件程序中的指令。

扩充内存(expanded memory)

可以由实模式程序使用的并且可以让扩充内存管理通过编页进行控制的640KB 以上的内存。

表达式(expression)

一个操作数或者多个操作数与操作符的组合。当对其运算时，将得到一个单一的值。

表达式列表(expression list)

一个表达式或者用逗号隔开的一组表达式。

扩展内存(extended memory)

在至少使用80286微处理器的计算机上，可以让保护模式的程序使用的640KB 或者1MB 以上的内存。

扩展名(extension)

参见文件扩展名。

域，字段(field)

(1) 一个结构中的一个成员。(2)记录中的数据单元。一个记录由一个或多个字段组成。一个或多个记录就组成一个表(数据库文件)。

文件句柄(file handle)

当文件打开时返回的整数值，所有的 I/O 操作都可以通过该值引用该文件。

文件类型(file type)

Basic 使用的文件的类型。文件扩展名通常可以指明文件的类别。例如，.FRM 表示窗体文件，.VBX 表示定制控件文件，.MAK 表示项目文件，而.TXT 表示文本文件或源代码文件等等。

文件名(filename)

用来引用一个文件的名称，由基本名和扩展名组成。可任意命名的基本名后面可以跟有一个句点及最多三个字符的文件扩展名。在 MS-DOS 和 Windows 3.1中，基本名最多可以有8个字符。在 Windows 95中，文件名可以更长而且可以包含空格。

文件扩展名(filename extension)

文件的扩展名，最多由三个字符组成而且前面要加一个句点与文件名隔开。文件扩展名通常代表文件类别。例如 VB32.EXE 中的扩展名.EXE 表示该文件是一个可执行文件。

过滤器(filter)

用来选取对象的某个子集进行处理的一组条件。

浮点(floating point)

浮点数据类型。存储增加了一组位数来表示的带有小数点的数字的技术。例如1.345E＋04。

字体(font)

给定的字样、字形和大小的完整字符集。

窗体(form)

由一个或多个页表格组成的 Visual Basic 包容器类。窗体的动作与窗口或对话框相似。

窗体页(form page)

窗体的某个单一页的 Visual Basic 包容器类。一个窗体页包含多个控件。在一个窗体中，只有一个窗体页可以是活动的。

窗体集(form set)

一个或多个相关的窗体的集合组成的 Visual Basic 包容器类。

函数(function)

执行某项任务的程序代码部分。Visual Basic 系统函数是内置在 Basic 中的。编程人员可以创建用户定义的函数(UDF)。因为函数需要零个或多个参数并且返回一个值，所以对函数的调用或引用可以是表达式的一部分。

函数调用(function call)

将控制传递给函数的语句。函数调用也传递参数。被调用函数向调用函数返回一个值。

全程变量(global variable)

在整个程序运行过程中都可视的变量。在 Visual Basic 中，在公共部分声明的变量是全程变量。

图形对象(graphic object)

报表或屏幕中使用的线或框。

十六进制(hexadecimal)

用字符0到9和 A 到 F 代表的16个基本数字。字母 A 到 F 对应于十进制的10到15。例如，3E9F 是一个十六进制数。

隐藏(hide)

在不禁止某个窗口的情况下使其消失。

热键(hotkey)

在菜单条或菜单选项中加下划线的字母，表示将该字母和 Alt 键一起按下来选取某个菜单选项。

I/O(输入/输出)

数据的交换。例如,键盘是一种输入设备,终端屏幕是一种输出设备。文件I/O表示读和写文件。

.ICO

图标资源文件的扩展名。

图标(icon)

图标是用户界面中的一个小图像,用来代表程序文件、数据文件或其他一些计算机实体。当最小化一个应用窗口时,就显示图标。

标识符(identifier)

变量、常数、宏、函数或数据类型的名称。该名称是由编程人员设计的。

递增(increment)

按顺序计数(例如6、7、8、9)。

下标,索引(index)

(1) Visual Basic 数组的下标。(2) 控制访问数据库文件的顺序的一个文件或复合索引标记。此顺序取决于关键字段。

继承(inheritance)

面向对象编程的术语,它表示子类的功能接受了其基于的类的特性。如果父类的特性变了,子类就继承变化过的特性。例如,如果用户向类中增加了新的属性,基于该类的所有子类就都可以使用这个新属性。

初始化(initialize)

当创建某个变量时为其赋值的过程。例如,x=3。

输入焦点(input focus)

被激活要接收输入的 Windows 对象的状态。在用户单击或按 Tab 键切换到某个应用的窗口对象后,该窗口就有输入焦点。

输入屏蔽(input mask)

控制用户输入的格式的属性。

插入模式(insert mode)

键入的所有字符都插入到光标位置的文本编辑状态,光标右侧的文本都向右移动。这种

模式与覆盖模式相对。

实例(instance)

面向对象编程的术语，它表示从某个类中创建的所有单个的对象。仅当对象的实例存在并且可以用于执行操作时，类才是对象的定义。

整数(integer)

没有分数或小数部分的数字。

不可见按钮(invisible button)

看不见的屏幕控件。它的操作过程由相应的程序定义。

键(key)

(1) 从键盘接收的代码。(2) 用来识别记录的字段或表达式。

键盘快捷键(keyboard shortcut key)

用户可以使用击键的组合而不是使用鼠标来选取菜单选项。例如，快捷键Ctrl+F可以选取Edit菜单的Find选项。

关键字(keyword)

Visual Basic语言的术语，它已有预先定义好的意义并向编译器提供指令。例如，If,Else和Declare都是关键字。

生命期(lifetime)

某个变量存在(可见)的阶段。私有变量的存在期从它被创建开始到从定义它的函数或过程返回为止。公共变量的存在期为从它被创建直到程序结束为止。

链接(link)

用在对象的链接与嵌入(OLE)中的术语，它表示源文件和目标文件之间的连接。链接的对象是对目标文件的对象的引用。当源文件变化时，目标对象也会变化。

连接器(linker)

组合起几个目标模块以生成一个可执行模块的软件程序。目标文件中含有相对的地址。这些目标模块可以是用户定义的或者从库中提取的模块。连接器将建立可执行模块所需的目标模块载入内存，然后将相对地址替换成绝对地址。上述处理将连续进行，直到所有的外部引用都消除为止。最后，连接器会创建一个可执行模块。

连接(linking)

连接器执行的操作。

本义字符(literal)

指定为某个格式的常数部分的字符，例如用于数据格式中的斜线。本义字符也用来指明某一类说明，象“本义字符文件名(literal filename)”，就表示用户可以直接输入实际的文件名而不是输入常数字符串。

本地数据(local data)

存在于客户机中的数据。

局部变量(local variable)

仅可在某个函数或过程内访问的变量。全程变量可以在函数和过程之间之外访问。

锁定(lock)

用来防止其他用户修改文件的一个处理。

逻辑操作符(logical operator)

组合起几个逻辑表达式以生成一个复合逻辑表达式的操作符。AND 和 OR 均为逻辑操作符。

循环(loop)

程序的一段重复结构。

低级语言(low-level language)

在复杂性上与机器代码类似的编程语言。汇编语言就是一种低级语言。

机器代码(machine code)

计算机使用的“母语”，它由计算机可以理解的执行程序的二进制计算机指令组成。

宏(macro)

(1)当按下一个键时出现的一组字符。

(2)同使用#define 编译器指令的表达式等效的常数标识符。在源文件编译之前，预处理器会用与标识符等效的字符来替换每一个出现标识符的地方。

宏代换(macro substitution)

在程序编译之前，用程序中的#define 编译器指令替换同表达式等效的常数标识符的活动。

主窗口(main Window)

在 Visual Basic 中，位于所有的系统窗口和用户定义的窗口之下的窗口。

主程序(main program)

程序执行开始的那个程序。

最大化(maximize)

使窗口扩大以充满整个屏幕或其父窗口。

中介模块(medium module)

带有一个数据块和一个或多个代码块的内存模块。

菜单(menu)

当从菜单条中选取菜单标题时出现的选项列表。它也指下拉式菜单。

菜单条(menu bar)

在窗口的顶部、标题条下面的一条,用来显示菜单。

菜单名称(menu name)

在菜单条上指示菜单的文本或图标。

菜单选项(menu option)

在被选取时执行一定的活动的菜单上的条目。菜单选项有时称为菜单命令,因为当选择一个菜单选项时会发生一些活动。

菜单块(menu pad)

菜单上带有菜单名的区域。

菜单系统(menu system)

菜单条、带有相关的下拉菜单的菜单块和下拉菜单选项的组合。

方法(method)

面向对象编程的术语,它表示对象可以执行的活动。

最小化(minimize)

将窗口缩小为图标。

模态的(modal)

窗口、窗体或对话框对象的一个特性。当一个模态窗口、窗体或对话框是活动的时候,在该模态对象消失或关闭之前任何其他窗口都不能激活。例如,错误框就是一个模态窗口。

非模态的(modeless)

窗口、窗体或对话框对象的一个特性。当一个非模态窗口或窗体是活动的时候，在不关闭非模态窗口之前，其他窗口均能激活。例如，Debug，Trace 和 Project 窗口都是非模态的。

多重实例(multiple instances)

从同一个类中创建的几个对象。每一个类都有自己的属性和私有数据，但是，所有的实例都共享相同的代码。

嵌套循环(nested loop)

一个循环嵌在另一个循环中的结构。

非模态的(nonmodal)

窗口或对话框的一个特性。当非模态窗口是活动的时候，允许其他窗口放到前面。例如，Debug 窗口就是一个非模态窗口。

空(null)

未赋值。空值不等效于空格或零或任何其他值。

空字符(null character)

带有零值 '\0' 的字符，它标志着字符串的结束。

对象(object)

类的一个实例，它包含数据和过程。

目标文件(object file)

包含目标代码的文件，它的扩展名为.OBJ。连接器将目标文件和库中的目标模块转换成可执行程序。

对象的链接与嵌入(OLE)

定义在两个应用程序之间共享数据的协议。对象，例如源(或对象)应用程序中的文档、图形或图片，可以链接或嵌入到目标(包容器)文档，例如报表中。

目标模块(object module)

编译器创建的目标文件。编译器将源程序转换成目标模块。

ODBC

参见开放式数据库连接。

OLE 自动化(OLE automation)

在程序中从一个应用程序控制另一个应用程序中的OLE的能力。

OLE 2.0控件(OLE 2.0 control)

预封装的OLE 2.0对象，它不同于其他OLE对象。它以与16位应用程序中的VBX控件同样的方式，用作32位应用程序中的控件。OLE控件有事件，并且可以成为子类来创建其他控件。OLE控件是OCX文件类型，它可以用Microsoft Visual C++ 2.1提供的Microsoft OLE Custum Control Developer's Kit来创建。

开放式数据库连接(ODBC，Open Database Connectivity 的缩写)

数据库服务器的标准协议。要将Basic与服务器相连并且访问远程数据，必须安装合适数据库服务器驱动程序。

操作数(operand)

表达式中操作的一个值。

操作符(operator)

指明表达式的一个或多个操作数如何运算的一个或多个字符。例如，表达式1+2中的+操作符代表将两个操作数相加。

优化(optimize)

使某些处理执行得尽可能高效率。

覆写模式(overtype mode)

与插入模式相反的一种文本编辑模式。当在覆写模式下键入字符时，当前光标处的字符被替换。

参数(parameter)

接收传递给函数的值的标识符。

按引用传递(pass by reference)

向用户定义的函数(UDF)中传递变量的一种方式。调用程序将变量的地址传递给UDF或过程。UDF或过程使用该地址检索或修改变量的值。数组就是按引用传递的。很显然，表达式总是按值传递的。按引用传递的结果是被传递的变量可以改变。

按值传递(pass by value)

向用户定义的函数(UDF)中传递变量的一种方式。在此方式下，将创建一个变量的拷贝并将其传递给UDF。UDF可以在不影响原始变量的情况下修改该拷贝。当UDF返回时，就舍弃该拷贝。参见按引用传递。

粘贴(paste)

在插入点处插入剪贴板中的内容。

路径名(pathname)

指定文件或目录的存储位置的文件全名。路径中可以包含驱动器名、目录名和文件名。例如,C:\VB4\SAMPLES\MYFILE.FRM。

象素(pixel)

屏幕或打印机可以显示的最小元素。

点(point)

(1)字体的度量单位。12点等于1 pica,6 pica(72点)约为1英寸。(2)将鼠标指针指向屏幕上的某个位置。

多态性(polymorphism)

面向对象编程的术语,这表示几个方法有同样的名称但是有不同的行为,这取决于使用方法的对象的类型。

指针(pointer)

在屏幕上随着鼠标移动而移动的一个箭头。

下拉式控件(pop-up control)

右侧的灰方框中有一个加下划线的箭头的矩形。单击箭头就会出现一个选项列表,用户可以从列表中选取一个选项。在 Visual Basic 中这叫做组合框控件。

下拉式菜单(pop-up menu)

当从对话框中选取下拉式控件时出现的一个选项列表。

优先权(precedence)

代表某个操作在过程表中占据的位置的一个术语。高优先权的操作符比低优先权的操作数先求值。

预处理器(preprocessor)

在编译器处理源代码之前,将源代码修改成与指令一致的处理器,它附属于编译器。

私有变量(private variable)

从被创建开始存在直到定义它的过程或函数将控制返回给调用程序为止的变量。私有变量在自己的过程中和所有被调用的过程中是可视的。

过程(procedure)

作为一个单元起作用的一组指令。一个过程可以执行另一个过程中的语句。后者执行自己的指令然后将控制返回给第一个过程。

编程语言(programming language)

用来编写程序的符号及相关的规则和语法的集合。

项目(project)

带有.MAK扩展名的一个特殊文件,它记录创建应用所需的全部文件。可以包含在项目中的文件类型有程序、屏幕、菜单、库和窗体文件。

属性(property)

控件对象的属性。用户可以设置一个属性来定义对象的一个特点或者对象的行为的一个方面。例如,Enabled 属性会影响某个控件对象在运行时间是否可以激活。

保护模式(protected mode)

X86(286、386、486、Pentium P5和Pentium P6)处理器家族的运行模式之一。在这种模式下,同时运行的程序不会侵占彼此的内存空间或者直接访问输入/输出设备,在多任务操作时,防止系统错误。另一种运行模式为实模式。Windows 的标准和增强运行模式是处于保护模式之下的。

伪代码(Pseudocode)

用来列出想让程序执行的活动的本质语言。

公共变量(public variable)

在所有的过程和函数中都可视的变量。它们用Global 命令定义。

纯虚函数(pure virtual function)

没有函数体的虚函数。它不需要函数体。

按式按钮(push button)

在Visual Basic 4.0的术语中,按式按钮是当被按下时要执行某些活动的一个控件。按式按钮是一个包含有文本或图形的矩形,同机械按钮相似。在Visual Basic 中,按式按钮指的是命令按钮。

单选按钮(radio button)

在Visual Basic 4.0的术语中,单选按钮是一个类似于圆形按钮的小圆形控件。一般情况下,会出现一个以上的单选钮。通常会选择一组单选按钮中的一个单选按钮,并在该按钮的小圆圈内标一个圆点符号。使用单选按钮,可以从几个相关的对象中作出选择。用户可以选

择另一个单选按钮，圆点符号会移动到新选择的单选按钮中。在 Visual Basic 中，单选按钮指的是选项按钮。

实模式(real mode)

X86(286,386,486,P5和 P6)处理器家族的运行模式之一。在这种模式下，会在内存中给程序一个确定的存储位置并且程序可以直接访问外围设备。另一种模式是保护模式。Windows 的实运行模式是处于实模式下的。

记录(record)

(1) 一种与结构等同的数据结构。(2) 数据库文件中的存储单元。表包含一个或多个记录，每个记录包含一个或多个字段。

递归函数(recursive function)

调用自身的函数。Visual Basic 支持递归的用户定义的函数。

关系表达式(relational expression)

用变量和关系操作符组成的值为布尔型的表达式。

关系操作符(relational operator)

比较两个操作数的符号(例如，=，<，>，==，>=和<=)。

远程数据(remote data)

存储在用 ODBC 可以访问的源中的数据。

返回值(return value)

用户定义的函数(UDF)返回给调用过程或 UDF 的值。

RGB 值(RGB values)

红、绿和蓝的值。RGB 值同 RGB()语句一起使用，并且可以用象@这样的命令指定。

例程(routine)

设计用来执行特定的有限任务的一组编程指令。它是子例程、过程或函数的术语。

运行阶段(run time)

程序执行阶段的时间。

科学记数法(scientific notation)

表示带有10的幂的数字值的方法，通常用于表示非常大或非常小的数字。例如，4.64002E+44表示4.64002乘以10的44次方。字母 E 表示后面是指数。

作用域(scope)

指某些变量的可见范围，即变量等标识符可以被访问的程序区域

滚动条(scroll bar)

用鼠标滚动窗口中的内容的控件。垂直滚动条在窗口的右侧，用来将内容上下滚动。水平滚动要在窗口的底部，用来将内容左右滚动。

选择(select)

在用户界面中，用鼠标或键盘指定在从菜单中选取条目后要发生什么活动。

选取框(selection marquee)

在窗体的设计中用来选取一组对象的点划线矩形。包容进框的所有对象均被选取。

服务器(server)

在网络上控制访问资源的计算机，资源有打印机、文件、数据库和调制解调器等。

符号(sign)

数字的正、负标志。

框架(skeleton)

指定搜索屏蔽的包含文本和通配符的字符串。例如，可以使用框架“F＊.FRM”来搜索以F开头的、扩展名为.FRM的所有文件。

源文件(source file)

包含Basic语言代码的文本文件。

源程序(source program)

包含编程语言的语句的程序版本。例如在Visual Basic中，可以将窗体文件转化成源代码文件。

SQL(结构化查询语言)

用来处理存储在表中的数据的标准化语言。

堆栈(stack)

数据存储的内存区域，并且数据按先入后出(FILO)的顺序检索。堆栈为函数调用和局部变量存储信息。

语句(statement)

同一个命令有关的一行或多行程序。

字符串(string)

一组文本字符。

字符串常量(string constant)

括在单引号('')、双引号("")或方括号([])中的字符串。字符串常量可以用在字符表达式中。例如,"any characters"和 'any characters' 都是字符串常量。

字符串文字(string literal)

字符串常量。

结构化查询语言(Structural Query Language (SQL))

参见 SQL。

子类(subclass)

使用另一类(父类)作为起点的新的类定义。子类继承了父类的特性。

风格(style)

字体的多样性,例如黑体、斜体或下划线,或者是上述风格的组合。

子例程(subroutine)

可以被其他程序调用的程序的一部分。

下标(subscript)

用来引用数组元素的一个整数。

语法(syntax)

管理编程语言的语句和命令结构的规则。

制表顺序(tab order)

当按 Tab 键时,输入焦点在控件之间移动的顺序。

任务(task)

要执行的工作,这是进度表中的基本单元。

终止符(terminator)

结束一个字符串的字符。

文本(text)

可阅读的字符,包括字母、数字和标点符号。

文本编辑窗口(text editing window)

用来创建或编辑程序的文本文件的 Basic 子系统。

文本文件(text file)

包含文本(ASCII 字符)的文件。

标题条(title bar)

在窗口的顶部显示窗口标题的区域。

开关选项(toggle option)

将某个条件打开或关闭的源代码指令。

工具栏(toolbar)

一组控件按钮。工具栏窗口可以浮动式放置。

工具栏提示(toolbar tips)

描述工具栏按钮的小文本窗口。当鼠标移到某个工具栏按钮上时就显示工具栏提示。

跟踪(trace)

一步一步地执行程序以找到软件错误的原因。

事务处理(transaction)

由完成指定的功能或活动的处理步骤组成的工作单元。

twip

与屏幕无关的测量单位。twip 是1/20点或者1/1440英寸。

类型(type)

数据的分类。例如,变量的数据类型可以是 Any 数据类型、货币型、双精度型、整型、长整型、单精度型、字符串型、用户定义类型和变体型。

类型检查(type checking)

编译器或译码器用来检查操作符的操作数是否为正确的数据类型的过程。因为 Basic 支持动态联合,译码器就执行类型检查。

超前输入缓冲区(typeahead buffer)

存放由于快速输入而比 Basic 需要的要多的那些字符的缓冲区。

UDF

参见用户定义的函数。

单目操作符(unary operator)

带有一个操作数的操作符。在 Basic 中，^ 和!不是单目操作符。当把＋和－用在一个数字或数字变量前面表示其正或负时，＋和－就是单目操作符。

解锁(unlock)

从文件上的删除锁定的操作，以便在多用户环境下该文件可以让其他用户修改。

用户定义的类(user-defined class (UDC))

由用户定义的类。这种类没有可视代表。

用户定义的函数(usser-defined function (UDF))

以 FUNCTION 命令开头，返回一个值并且可以在表达式中调用的过程。

有效性验证(validation)

检查插入或修改的数据的操作，以确保其满足一定的条件和限制。

变量(variable)

在内存中存放数据的位置，它按标识符来引用。

可视性(visibility)

从某个应用程序中的多个位置引用(或观察)变量或函数的能力。可视性这个术语是用来指定变量或函数可以按名称引用的程序的范围。例如，局部变量的可视性仅限于创建它的过程中。

通配符(wildcard)

当不知道文件名时，?和 * 字符可以用来形成一个搜索文件的模板。?通配符代表任意单个字符，* 通配符代表文件名中的任何多个字符。在 LIKE 函数中，* 通配符代表连续的零个或多个字符。

窗口(window)

显示输出和接收输入的屏幕上的矩形区域。

向导(wizard)

指示用户输入执行某些任务所需的信息的一系列对话框。当向导有了其所需的全部信

息，它就完成了指定的任务。

单词折行（word wrap）

文本编辑器执行的一个处理。在文本编辑中当某些行长超过了编辑窗口的右边界时，该行就在编辑窗口可以显示的最后一个单词后面被截断，该行的其余部分显示在下一行中。Basic 的文本编辑器有一个选项，可以让用户将单词折行模式置为开或关。

缩放（zoom）

将窗口扩大以充满屏幕。如果窗口已经扩大，它就缩小为原来的大小。

780
860
840
40
0
0
4 0
131

38
45
1900

50000.
3000.
4160.
7160
1900
9060

570
300
870

600

54160
5000
54160
3000
57160
699
699
26
785
19
1265
587
1587
38
5
248
107
1
349
12
371
1
383
392
215
6724
699
699

新华书店哈尔滨发行所

搭上轻松快车

与最新流行软件同行

“轻松学习最新流行软件系列丛书”
的主旨是以最快的方式
轻松易学的形式介绍当今国内外
最著名软件公司的重要产品
使读者可以快速通览当前流行软件领域的全貌
迅速掌握最新软件的使用方法和编程技术
轻轻松松从新手变为高手

本系列丛书有以下特点

阵容強大

本丛书內容涉及
Microsoft、Novell、Borland、AutoDesk、Adobe
等当前最知名软件公司的流行产品
尤以中文本地化产品为介绍的重中之重
同时密切结合软件领域的技术发展
配合最新流行软件的推出
不断推出配套新书

版本先进

紧跟最新的软件版本
使丛书的出版
与最新软件版本的更新保持同步

內容独到

结合应用的实际
着重技术精华的剖析和操作技巧的指点 明晰实用
既可作为教材，循序渐进学习
亦可作为上机伴侣，随用随查
使读者能充分融汇软件的奥妙，举一反三

风格新颖

配有设计新颖的內容提示图标
别具一格的內容导读索引
大量生动的屏幕显示画面
使读者如临操作现场
充分体味软件学习之快乐

轻松学习最新流行软件系列丛书

1、轻松学习 Windows 95 中文版
2、轻松学习 Word for Windows 95 中文版
3、轻松学习 Excel for Windows 95 中文版
4、轻松学习 Office for Windows 95 中文版
5、轻松学习 PowerPoint for Windows 95
6、轻松学习 Visual C++ 4.0
7、轻松学习 Visual Basic 4.0
8、轻松学习 Visual FoxPro 3.0
9、轻松学习 Borland C++ 5.0
10、轻松学习 Delphi for Windows

ISBN 7-302-02114-7

定价：25.00 元

责任编辑 黄娟娟　　封面设计 葛占基